A não tão pequena introdução ao $\LaTeX\,2_\varepsilon$

Ou $\LaTeX\,2_\varepsilon$ em 118 minutos

por Tobias Oetiker

Hubert Partl, Irene Hyna e Elisabeth Schlegl

Tradução portuguesa por Alberto Simões

Versão 4.00, 12 de Dezembro de 2002

Obrigado!

Muito do material utilizado nesta introdução provém de uma introdução austríaca ao LaTeX 2.09 escrita em alemão por:

Hubert Partl `<partl@mail.boku.ac.at>`
Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna `<Irene.Hyna@bmwf.ac.at>`
Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl `<no email>`
in Graz

Se está interessado no documento alemão, pode encontrar a versão actualizada para LaTeX 2_{ε} por Jörg Knappen a partir de
`CTAN:/tex-archive/info/lshort/german`

Durante a preparação deste documento, pedi comentários em `comp.text.tex`. Recebi bastantes respostas. Os indivíduos seguintes ajudaram com correcções, sugestões e material para melhorar este documento. Colocaram grande empenho para me ajudar a colocar este documento na sua presente forma. Gostaria de agradecer sinceramente a todos eles. Naturalmente, todos os erros que encontrar neste livro são meus. Se encontrar alguma palavra que esteja escrita correctamente, deve ter sido uma das pessoas abaixo que me enviou essa linha.

Rosemary Bailey, Marc Bevand, Friedemann Brauer, Jan Busa,
Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos,
Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke,
Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes,
Matthias Dreier, David Dureisseix, Elliot, Hans Ehrbar, Daniel Flipo, David Frey,
Hans Fugal, Robin Fairbairns, Jörg Fischer, Erik Frisk, Mic Milic Frederickx,
Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Cyril Goutte,
Greg Gamble, Neil Hammond, Rasmus Borup Hansen, Joseph Hilferty, Björn
Hvittfeldt, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey,
Byron Jones, David Jones, Johannes-Maria Kaltenbach, Michael Koundouros,
Andrzej Kawalec, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo,
Maik Lehradt, Rémi Letot, Johan Lundberg, Alexander Mai, Martin Maechler,
Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren,
Lenimar Nunes de Andrade, Demerson Andre Polli, Maksym Polyakov
Hubert Partl, John Refling, Mike Ressler, Brian Ripley, Young U. Ryu,
Bernd Rosenlecher, Chris Rowley, Risto Saarelma, Hanspeter Schmid,
Craig Schlenter, Baron Schwartz, Christopher Sawtell, Geoffrey Swindale,
Boris Tobotras, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli,
Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone,
e Mikhail Zotov.

Prefácio

O LaTeX [1] é um sistema tipográfico, bastante adequado para produzir documentos científicos e matemáticos de grande qualidade tipográfica. O sistema é igualmente adequado para produzir todo o tipo de outros documentos, desde simples cartas até livros completos. O LaTeX usa o TeX [2] como sistema de formatação.

Esta pequena introdução descreve o LaTeX 2_ε, e deve ser suficiente para um grande número das aplicações do LaTeX. Consulte [1, 3] para uma descrição completa do sistema LaTeX.

O LaTeX está disponível para quase todos os computadores, desde o PC e Mac até grandes sistemas UNIX e VMS. Em muitas redes de computadores de universitários, irá descobrir que uma instalação do LaTeX está disponível e pronta a ser utilizada. Informações sobre como aceder à sua instalação local do LaTeX deve estar incluída no *Local Guide* [4]. Se tiver problemas quando começar a trabalhar, pergunte à pessoa que lhe deu este livro. O âmbito deste documento *não* é explicar como instalar e preparar o LaTeX, mas ensinar a escrever os seus documentos de forma a que possam ser processados pelo LaTeX.

Esta introdução está dividida em 5 capítulos:

O Capítulo 1 indica a estrutura básica de um documento LaTeX 2_ε. Aprenderá também um pouco da história do LaTeX. Após a leitura deste capítulo, deverá ter uma ideia básica do funcionamento do LaTeX. Esta ideia será uma mera estrutura, mas irá permitir integrar a informação dada nos capítulos seguintes.

O Capítulo 2 apresenta com algum detalhe como escrever um documento. Explica igualmente a maior parte dos comandos e ambientes essenciais do LaTeX. Após a leitura deste capítulo, estará pronto a escrever o seu primeiro documento.

O Capítulo 3 explica como escrever fórmulas com o LaTeX. Mais uma vez, numerosos exemplos ajudarão a perceber como usar uma das maiores potencialidades do LaTeX. No final deste capítulo, encontrará tabelas com listas de todos os símbolos matemáticos disponíveis em LaTeX.

O Capítulo 4 explica a geração de índices e bibliografias, inclusão de gráficos EPS, e algumas outras extensões úteis.

O Capítulo 5 contém alguma informação potencialmente perigosa sobre como fazer alterações aos formatos *standard* produzidos pelo LaTeX. Mostrará como modificar coisas de modo que o belo resultado do LaTeX comece a ficar bastante mau.

É importante ler os capítulos sequencialmente—afinal o livro não é assim tão grande. Leia cuidadosamente os exemplos, porque grande parte da informação está contida nos vários exemplos que vai encontrar ao longo do livro.

Se precisar de mais algum material relacionado com o LaTeX dê uma vista de olhos a um dos arquivos ftp do *Comprehensive TeX Archive Network* (`CTAN`). Podem ser encontrados, por exemplo, em `ctan.tug.org` (US), `ftp.dante.de` (Alemanha), `ftp.tex.ac.uk` (UK). Se não está num destes países, procure um arquivo perto de si.

Encontrará outras referências ao CTAN ao longo deste livro. Especialmente, apontadores para *software* e documentos que poderá querer copiar para o seu computador. Em vez de escrever endereços completos, escreveremos apenas `CTAN:` seguido da localização que deve visitar dentro da árvore do CTAN.

Se deseja instalar o LaTeX para utilização no seu computador, visite `CTAN:/tex-archive/systems`.

Se tiver ideias sobre alguma coisa que deva ser adicionada, alterada ou removida deste documento, por favor, avise-me. Estou interessado especialmente em respostas dos que estão a aprender LaTeX sobre quais as partes desta introdução fáceis de compreender e quais as que devem ser explicadas melhor.

Tobias Oetiker `<oetiker@ee.ethz.ch>`

Departamento de Tecnologia da Informação e
Engenharia Electrotécnica,
Instituto de Tecnologia Federal da Suíça

A versão actual e original deste documento está disponível em `CTAN:/tex-archive/info/lshort` A versão portuguesa está disponível em `CVS` em `http://natura.di.uminho.pt`. Tradução por
Alberto Simões <*albie@alfarrabio.di.uminho.pt*>
Departamento de Informática da Universidade do Minho

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Coisas Que Precisa de Saber

Na primeira parte deste capítulo, será apresentada uma visão geral sobre a filosofia e história do LaTeX 2_ε. A segunda parte foca as estruturas básicas de um documento LaTeX. Depois de ler este capítulo, terá uma ideia geral de como o LaTeX funciona. O seu principal objectivo é ajudar a integrar toda a informação que é dada nos capítulos seguintes.

1.1 O Nome do Jogo

1.1.1 TeX

O TeX é um programa de computador criado por Donald E. Knuth [2]. É devoto à tipografia de texto e formulas matemáticas. O motor tipográfico começou a ser escrito em 1977 para explorar os potenciais de equipamento digital de impressão que estava a infiltrar na indústria de publicação naquele tempo, especialmente na esperança de poder alterar o rumo da deterioração de qualidade tipográfica que ele viu a afectar os seus próprios livros e artigos. Tal como o usamos hoje, o TeX foi disponibilizado em 1982 com pequenos melhoramentos adicionados em 1989 para suportar da melhor forma os caracteres de 8-bits, e múltiplas linguagens. Tem recebido muitos elogios por ser extremamente estável, funcionar em muitos tipos diferentes de computadores, e virtualmente não ter qualquer *bug*. A versão do TeX está a convergir para π e de momento é 3.14159.

Pronunciamos TeX como "Tech," com um "ch" idêntico à palavra alemã "Ach" ou a escocesa "Loch." Em ambientes ASCII, TeX deve ser escrito como `TeX`.

1.1.2 LaTeX

O LaTeX é um pacote de comandos (*macros* que permite que formatos predefinidos, de grandíssima qualidade tipográfica, sejam impressos por qualquer autor. Foi escrito originalmente por Leslie Lamport [1]. Usa o TeX

como o seu motor tipográfico. Ultimamente, o LaTeX é mantido por Frank Mittelbach.

LaTeX é pronunciado "Lay-tech" ou "Lah-tech." Se se referir ao LaTeX num ambiente `ASCII`, deve escrever `LaTeX`. LaTeX 2_ε é pronunciado "Lay-tech two e" e escrito `LaTeX2e`.

1.2 Bases

1.2.1 Autor, Paginadores e Tipógrafo

Para publicar alguma coisa, os autores dão um manuscrito dactilografado à companhia de publicação. Um dos seus paginadores decide o formato do documento (largura da coluna, tipos de letra, espaços antes e após os cabeçalhos, ...). Este escreve as suas instruções no manuscrito, que é entregue ao tipógrafo que imprime o livro de acordo com estas instruções.

Um paginador humano tenta imaginar o que o autor tinha em mente enquanto escrevia o manuscrito. Tem de decidir os cabeçalhos de capítulos, exemplos, formulas, etc. baseado nos seus conhecimentos profissionais e no conteúdo do manuscrito.

Num ambiente LaTeX, o paginador é o LaTeX, que usa o TeX como seu tipógrafo. Mas o LaTeX é "apenas" um programa e portanto precisa de algum apoio. O autor tem de dar informação adicional descrevendo a estrutura lógica do seu trabalho. Esta informação é escrita no texto como "comandos LaTeX."

Esta é uma grande diferença da abordagem WYSIWYG[1] que a maior parte dos processadores de texto modernos, tais como o *MS Word* ou *Corel WordPerfect*, usam. Com estas aplicações, os autores especificam o formato do documento de uma forma interactiva enquanto vão escrevendo o texto no computador. Durante esse processo, podem ver no écran como o resultado final vai aparecer quando impresso.

Ao usar o LaTeX, normalmente não é possível ver o resultado final enquanto se digita o texto. No entanto, o resultado final pode ser pré-visualizado no écran depois de processar o ficheiro com o LaTeX. Então, podem ser feitas correcções para enviar posteriormente o documento para a impressora.

1.2.2 Desenho do Formato

O desenho relacionado com a tipografia é bastante trabalhoso. Autores não qualificados cometem frequentemente erros sérios de formatação assumindo que o desenho de livros é, na sua maior parte, uma questão estética — "Se um documento é artisticamente bonito, então está bem desenhado." Mas, como um documento tem de ler lido e não pendurado numa galeria de pinturas,

[1] What you see is what you get — O que vês é o que recebes.

a leitura e compreensão é de muito maior importância do que a sua forma. Exemplos:

- O tamanho e a numeração dos cabeçalhos devem ser escolhidos para fazer a estrutura de capítulos e secções clara ao leitor.
- O comprimento das linhas tem de ser suficientemente curto para não fazer o leitor trocar os olhos, mas suficientemente longo para preencher uma página de uma forma esteticamente bela.

Com sistemas WYSIWYG, os autores criam documentos esteticamente agradáveis sem estrutura, ou em que esta se apresenta de uma forma inconsistente. O LaTeX previne este tipo de erros de formatação ao obrigar os autores a declarar a estrutura *lógica* do seu documento, escolhendo depois o formato mais adequado.

1.2.3 Vantagens e Desvantagens

Quando as pessoas do mundo WYSIWYG conhecem pessoas que usam LaTeX, frequentemente discutem "as vantagens do LaTeX em relação a um processador de texto normal" ou o contrário. A melhor coisa que se pode fazer quando uma discussão inicia é manter a calma, porque este tipo de discussões saem facilmente dos limites. Mas por vezes não se pode escapar ...

Então, aqui estão algumas munições. As principais vantagens do LaTeX sobre um processador de texto "normal" são as seguintes:

- Formatos criados profissionalmente estão disponíveis, que fazem que um documento pareça realmente impresso numa tipografia;
- A escrita de formulas matemáticas é suportada de uma forma conveniente;
- O utilizador apenas precisa de aprender uma dúzia de comandos facilmente compreensíveis que especificam a estrutura lógica de um documento. Quase nunca se precisa de atormentar com o formato real do documento;
- Até estruturas complexas, tais como notas de rodapé, referências, tabelas de conteúdos e bibliografias podem ser facilmente geradas;
- Pacotes gratuitos podem ser aplicados a tarefas tipográficas não suportadas pelo LaTeX básico. Por exemplo, existem pacotes para incluir gráficos PostScript e para imprimir bibliografias conforme os *standards*. Muitos destes pacotes estão descritos no *The LaTeX Companion* [3];

- O LaTeX encoraja os autores a escrever textos bem estruturados porque é assim que o LaTeX funciona — especificando a estrutura;

- TeX, o motor de formatação do LaTeX 2_ε, é extremamente portável e livre. Desta forma, o sistema funciona em quase todas as plataformas de *hardware* existentes.

O LaTeX também tem algumas desvantagens, e penso que é um pouco difícil para mim encontrar alguma, mas parece-me que outras pessoas podem indicar centenas `;-)`

- LaTeX não funciona bem para pessoas que tenham vendido a sua alma. . .

- Apesar de alguns parâmetros poderem ser ajustados num formato pré-definido, o desenho de todo um novo formato é difícil e demora muito tempo [2]

- É difícil de escrever documentos mal estruturados e desorganizados;

- Embora o seu hamster demonstre alguma dedicação durante os primeiros passos, nunca conseguirá digerir completamente o conceito de marcação lógica.

1.3 Ficheiros LaTeX

Os ficheiros LaTeX são ficheiros de texto `ASCII` planos. Pode criá-los em qualquer editor de texto; o ficheiro contém o texto do documento assim como os comandos que dizem ao LaTeX como formatar o texto.

1.3.1 Espaços

Caracteres "brancos" como espaços ou caracteres de tabulação (*tabs*) são tratados uniformemente como "espaços" pelo LaTeX. Caracteres brancos *consecutivos* são tratados como *um* "espaço". Os espaços no início de uma linha são geralmente ignorados, e uma simples mudança e linha é tratada da mesma forma que um espaço.

Uma linha em branco entre duas linhas de texto define o fim de um parágrafo. *Várias* linhas vazias são tratadas da mesma forma que *uma* linha vazia. O texto que se segue é um exemplo. Do lado esquerdo apresenta-se o texto do ficheiro a escrever, e do lado direito o resultado depois de processado.

[2] Rumores dizem que este é um dos pontos chave a ser abordado na versão 3 do LaTeX.

```
Não interessa se introduz apenas
um ou vários     espaços depois
de uma palavra.

Uma linha em branco inicia um novo
parágrafo.
```

Não interessa se introduz apenas um ou vários espaços depois de uma palavra.

Uma linha em branco inicia um novo parágrafo.

1.3.2 Caracteres Especiais

Os símbolos que se seguem são caracteres reservados que ou têm um significado especial para o LaTeX ou não estão disponíveis em todos os tipos de letras. Se os introduzir directamente no seu texto, não serão impressos, mas causarão o LaTeX a fazer coisas que não deseja.

```
# $ % ^ & _ { } ~ \
```

Como irá ver, estes caracteres podem ser utilizados nos seus documentos todos da mesma forma, adicionando uma barra invertida como prefixo:

```
\$ \& \% \# \_ \{ \}
```

$ & % # _ { }

Outros símbolos e muitos mais podem ser impressos com comandos especiais em formulas matemáticas ou como acentos. O caracter de barra invertida (\) *não* deve ser introduzido adicionando uma outra barra antes (\\), porque esta sequência é utilizada para quebrar linhas de texto.[3]

1.3.3 Comandos LaTeX

Os comandos LaTeX são sensíveis às maiúsculas/minúsculas, e têm um de dois formatos:

- Começam com um backslash[4] (\) e têm um nome que consiste apenas de letras. Os nomes de comandos terminam com um espaço, um número ou qualquer outro símbolo "não-letra".
- Consistem num *backslash* e exactamente um caracter não letra.

O LaTeX ignora espaços em branco após os comandos. Se deseja um espaço depois de um comando, deve colocar `{}` e um caracter especial, ou um comando de espaçamento especial depois do nome do comando. O `{}` faz com que o LaTeX pare de comer todos os espaços após o nome do comando.

[3] Tente o comando `$\backslash$` em vez da dupla barra, que produz um '\'.
[4] NT: a partir de agora utilizaremos o termo inglês em vez de *barra invertida*

```
Li que o Knuth divide as
pessoas que trabalham com o \TeX{}
em \TeX{}nicos e \TeX pertos.\\
Hoje é \today.
```

Li que o Knuth divide as pessoas que trabalham com o TEX em TEXnicos e TEXpertos.
Hoje é 12 de Dezembro de 2002.

Alguns comandos precisam de um parâmetro que deve ser introduzido entre chavetas (`{}`) depois do nome do comando. Outros suportam parâmetros opcionais que são adicionados depois do nome do comando entre parêntesis rectos (`[]`). Os exemplos seguintes utilizam alguns comandos LATEX. Não se preocupe com eles, pois serão explicados mais tarde.

```
Pode \textsl{apoiar-se} em mim!
```

Pode *apoiar-se* em mim!

```
Por favor, comece uma nova linha
exactamente aqui!\newline
Obrigado!
```

Por favor, comece uma nova linha exactamente aqui!
Obrigado!

1.3.4 Comentários

Quando o LATEX encontra um caracter `%` ao processar o ficheiro, ignora todo o resto dessa linha, a mudança de linha e todos os espaços no início da linha seguinte.

Pode ser utilizado para escrever notas nos ficheiros de código, que não aparecerão nas versões impressas.

```
Este é um % estúpido
% Melhor: instrutivo <----
exemplo: Supercal%
              ifragilist%
    icexpialidocious
```

Este é um exemplo: Supercalifragilisticexpialidocious

O caracter `%` também pode ser utilizado para quebrar linhas longas onde não são permitidos espaços nem mudanças de linha.

Para comentários mais longos, pode utilizar o ambiente `comment` disponibilizado pelo pacote `verbatim`. Isto significa que, para usar o ambiente `comment` tem de adicionar o comando `\usepackage{verbatim}` ao preâmbulo do seu documento.

```
Este é outro
\begin{comment}
bastante estúpido,
mas instrutivo
\end{comment}
exemplo de como embeber
comentários nos seus documentos.
```

Este é outro exemplo de como embeber comentários nos seus documentos.

Note que isto não funciona dentro de ambientes complexos tal como o matemático.

1.4 Estrutura do Ficheiro

Quando o LaTeX 2_ε processa um ficheiro, espera que ele respeite uma certa estrutura. Desta forma, todos os ficheiros devem começar com o comando

```
\documentclass{...}
```

Isto especifica o tipo de documento que tenciona escrever. Em seguida, pode incluir comandos que influenciam o estilo de todo o documento, ou pode incluir pacotes que adicionarão novas propriedades ao sistema LaTeX. Para incluir este tipo de pacote, utilize o comando

```
\usepackage{...}
```

Quando todo o trabalho de configuração estiver feito,[5] inicie o corpo do texto com o comando

```
\begin{document}
```

Agora introduza o texto misturado com comandos LaTeX úteis. No fim do documento adicione o comando

```
\end{document}
```

que indica ao LaTeX o fim do seu serviço. Tudo o que siga este comando será ignorado pelo LaTeX.

A figura 1.1 mostra o conteúdo de um ficheiro LaTeX 2_ε mínimo. Um ficheiro um pouco mais complexo é mostrado na figura 1.2.

1.5 Uma Sessão de Edição Típica

Aposto que deve estar a morrer por poder testar o pequeno exemplo de LaTeX mostrado na página 8. Aqui está alguma ajuda: o LaTeX, por si só, não contém um interface gráfico, ou botões para clicar. É apenas um programa que mastiga o seu documento. Algumas instalações do LaTeX providenciam

[5] A área entre `\documentclass` e`\begin{document}` é chamado de *preâmbulo*.

```
\documentclass{article}
\begin{document}
Pequeno é belo.
\end{document}
```

Figura 1.1: Um ficheiro LaTeX mínimo.

```
\documentclass[a4paper,11pt]{article}
% Define o título
\author{H.~Partl}
\title{Minimalista}
\begin{document}
% Gera o título
\maketitle
% Insere a tabela de conteúdos
\tableofcontents
\section{Algumas Palavras Interessantes}
Bem, e aqui está o inicio do meu adorado artigo.
\section{Adeus, Mundo!}
\ldots{} e aqui ele acaba.
\end{document}
```

Figura 1.2: Exemplo de um Artigo Real.

um programa gráfico para interagir com o LaTeX clicando em certos menus. Mas, Homens a sério não Clicam, portanto aqui está como fazer com que o LaTeX compile o seu documento num sistema baseado em consola. Note que esta descrição assume que está existe uma instalação LaTeX a funcionar no seu computador (Homens a sério usam Unix).

1. Edite/Crie o seu documento LaTeX. Este ficheiro deve ser texto ASCII sem qualquer tipo de formatação. Em Unix, todos os editores vão fazer o que deseja. Em Windows, deve ter a certeza que vai gravar o ficheiro em ASCII ou em formato *plano*. Ao escolher um nome para o seu ficheiro, tenha a certeza de lhe colocar a extensão `.tex`.

2. Execute o LaTeX sobre o seu ficheiro. Se correr tudo bem, irá acabar com um ficheiro `,dvi`. Poderá ser necessário correr o LaTeX várias vezes para obter um índice de conteúdos e todas as referencias internas correctamente. Sempre que o seu ficheiro tenha um erro o LaTeX irá dizer-lhe que erro cometeu e irá parar o seu processamento. Pressione `ctrl-D` para voltar à linha de comandos.

   ```
   latex foo.tex
   ```

3. Agora pode ver o ficheiro DVI. Existem várias formas de o fazer. Pode mostrar o ficheiro no écran com

   ```
   xdvi foo.dvi &
   ```

   Isto funciona apenas em Unix com X11. Se está em Windows, poderá tentar usar o `yap` (ainda outro pré-visualizador—yet another previewer). Também pode converter o ficheiro dvi em PostScript para imprimir ou ver com o Ghostscript.

   ```
   dvips -Pcmz foo.dvi -o foo.ps
   ```

   Se tiver sorte, o seu sistema LaTeX até trás a ferramenta `dvipdf`, que lhe permite converter o seu ficheiro `.dvi` directamente em pdf.

   ```
   dvipdf foo.dvi
   ```

1.6 O Formato do Documento

1.6.1 Classes de Documentos

A primeira informação que o LaTeX precisa de saber quando processa um ficheiro é o tipo de documento que o autor quer criar. Este tipo é especificado

pelo comando `\documentclass`.

`\documentclass[`*opções*`]{`*classe*`}`

Aqui, *classe* especifica o tipo do documento a ser criado. A tabela 1.1 lista as classes de documento explicadas nesta introdução. A distribuição do LaTeX 2ε permite classes adicionais para outros documentos, incluindo cartas e slides. O parâmetro *opções* caracteriza o comportamento da classe do documento. As opções devem ser separadas por vírgulas. As opções mais comuns para os documentos padrão estão listadas na tabela 1.2.

Exemplo: Um ficheiro para um documento LaTeX pode começar com a linha

```
\documentclass[11pt,twoside,a4paper]{article}
```

que instruí o LaTeX a escrever o documento como sendo um artigo (*article*) com um tamanho base de letra de onze pontos (*11pt*), e para gerar um formato adequado para impressão de ambos os lados (*twoside*) em papel A4 (*a4paper*).

1.6.2 Pacotes

Ao escrever o seu documento, provavelmente encontrará algumas áreas onde o LaTeX básico não conseguirá resolver os seus problemas. Se deseja incluir gráficos, texto colorido ou código fonte de um ficheiro no seu documento, irá precisar de melhorar as capacidades do LaTeX. Este tipo de melhoramentos

Tabela 1.1: Classes de Documentos.

`article` para artigos em jornais científicos, pequenos relatórios, documentação de programas, convites, ...

`report` para relatórios mais longos contendo vários capítulos, pequenos livros, teses de doutoramento, ...

`book` para livros verdadeiros

`slides` para slides. Esta classe usa letras grandes do tipo sans serif. Deve considerar utilizar o pacote FoilTeX[a] em vez do slides.

[a] `CTAN:/tex-archive/macros/latex/contrib/supported/foiltex`

Tabela 1.2: Opções das Classes de Documentos.

`10pt`, `11pt`, `12pt`	Define o tamanho principal das letras do documento. Caso não especifique uma delas, é assumido `10pt`.
`a4paper`, `letterpaper`, ...	Define o tamanho do papel. Por omissão, é utilizado o `letterpaper`. Além destes, existem `a5paper`, `b5paper`, `executivepaper`, e `legalpaper`. ,
`fleqn`	Alinha as formulas à esquerda em vez de as centrar.
`leqno`	Coloca a numeração nas formulas do lado esquerdo em vez do lado direito.
`titlepage`, `notitlepage`	Especifica se se deve criar uma nova página depois do título do documento ou não. A classe `article` não inicia uma nova página por omissão, enquanto que o `report` e o `book` o fazem.
`twocolumn`	Instruí o LaTeX a escrever o documento em duas colunas.
`twoside, oneside`	Indica se deve ser gerado resultado para impressão dos dois lados. As classes `article` e `report` são impressas apenas de um lado e a classe `book` é impressa dos dois lados por omissão. Note que esta opção muda apenas o estilo do documento. A opção `twoside` *não* diz à impressora que deve fazer a impressão de ambos os lados.
`openright, openany`	Faz os capítulos começar apenas nas páginas do lado direito ou na próxima disponível. Esta opção não funciona com a classe `article` uma vez que ela não sabe o que são capítulos. A classe `report` inicia por omissão os capítulos na primeira página disponível e a classe `book` inicia nas páginas do lado direito.
`landscape`	Muda o formato do documento para ser impresso numa folha deitada.

são chamados pacotes (*packages*), e são activados com o comando

`\usepackage[`*opções*`]{`*pacote*`}`

onde *pacote* é o nome do pacote e *opções* é uma lista de palavras chave que activam (ou desactivam) propriedades especiais no pacote. Alguns pacotes vêm com a distribuição base do LaTeX 2ε (Veja a tabela 1.3), outros são distribuidos separadamente. Poderá encontrar mais informação sobre os pacotes instalados no *Local Guide* [4]. A primeira fonte de informação sobre os pacotes LaTeX é o *The LaTeX Companion* [3], que contém descrições de centenas de pacotes juntamente com informação sobre como escrever as suas próprias extensões para o LaTeX 2ε.

1.7 Ficheiros que provavelmente encontrará

Quando trabalhar com o LaTeX, rapidamente se encontrará num labirinto de ficheiros com várias extensões e provavelmente sem pista alguma sobre o seu conteúdo. Em seguida apresentamos uma lista indicando os vários tipos de ficheiros que possivelmente encontrará ao trabalhar com TeX. Note que esta tabela não pretende ser uma lista completa de extensões, mas se encontrar uma que falte e que lhe pareça importante, avise!

`.tex` Ficheiros LaTeX ou TeX. Podem ser compilados com o `latex`.

`.sty` Pacote de macros LaTeX. Este é um ficheiro que pode carregar no seu documento LaTeX utilizando o comando `\usepackage`.

`.dtx` TeX documentado. Este é o formato base para a distribuição de ficheiros de estilo LaTeX. Se processar um destes ficheiros ficará com documentação do pacote contido no ficheiro .dtx.

`.ins` É o instalador para os ficheiros contidos nos ficheiros .dtx com o mesmo nome. Se fizer *download* de um pacote LaTeX da Internet, normalmente ficará com um ficheiro .dtx e um .ins. Corra o LaTeX no ficheiro .ins para descompactar o ficheiro .dtx.

`.cls` Ficheiros de classe que definem como o seu documento vai aparecer. Podem ser seleccionados com o comando `\documentclass`.

`.fd` Ficheiros de descrição de tipos de letra que comunica ao LaTeX novos tipos disponíveis.

Os ficheiros seguintes são gerados quando corre o LaTeX no seu documento:

Tabela 1.3: Alguns Pacotes Distribuídos com o LaTeX.

---

doc Permite a documentação de programas LaTeX.
Descrito em `doc.dtx`[a] e no *The LaTeX Companion* [3].

exscale Fornece versões das letras do modo matemático reguláveis em tamanho
Descrito em `ltexscale.dtx`.

fontenc Especifica que codificação de caracteres o LaTeX deve usar.
Descrito em `ltoutenc.dtx`.

ifthen Fornece comandos da forma
'if. . . then do. . . otherwise do. . . .'
Descrito em `ifthen.dtx` e no *The LaTeX Companion* [3].

latexsym Permite aceder ao tipo de letra *symbol* do LaTeX. Descrito em `latexsym.dtx` e no *The LaTeX Companion* [3].

makeidx Fornece comandos para produzir índices. Descrito na secção 4.3 e no *The LaTeX Companion* [3].

syntonly Processa um documento sem escrever o resultado de processamento.

inputenc Permite a especificação de uma codificação como seja ASCII, ISO Latin-1, ISO Latin-2, páginas de código 437/850 IBM, Apple Macintosh, Next, ANSI-Windows ou definido pelo utilizador. Descrito em `inputenc.dtx`.

---

[a] Este ficheiro deve estar instalado no seu sistema, e deve conseguir convertê-lo para um ficheiro `dvi` digitando `latex doc.dtx` em qualquer directoria em que tiver permissões. Isto também é verdade para todos os outros ficheiros mencionados nesta tabela.

`.dvi` Device Independent file. Este é o resultado principal da utilização do LaTeX. Pode ver o seu conteúdo com um programa de visualização de DVI ou pode enviá-lo para a impressora com `dvips` ou uma aplicação similar.

`.log` Contém informação detalhada sobre o que aconteceu da última vez que correu o LaTeX no seu documento.

`.toc` Guarda todos os cabeçalhos. Será lido da próxima vez que correr o LaTeX para produzir a tabela de conteúdos.

`.lof` Idêntico ao .toc mas para a lista de figuras.

`.lot` Mais uma vez o mesmo para a lista de tabelas.

`.aux` Outro ficheiro que transporta informação de uma utilização do LaTeX até à próxima. Entre outras coisas, o ficheiro .aux é utilizado para guardar informação associada com referências cruzadas.

`.idx` Se o seu documento incluí um índice, o LaTeX guarda todas as palavras que vão para o índice neste ficheiro. Este ficheiro deve ser processado com o `makeindex`. Visite a secção 4.3 na página 70 para mais informação sobre indexação.

`.ind` É o ficheiro processado do .idx, pronto para inclusão no seu documento no próximo ciclo de compilação.

`.ilg` Ficheiro com os resultados sobre o processamento do `makeindex`.

1.7.1 Estilo de Páginas

O LaTeX suporta três combinações pré-definidas de cabeçalho/rodapé — chamados de estilo de páginas . O parâmetro *estilo* pode ser um dos definidos pelo comando

`\pagestyle{`*estilo*`}`

A tabela 1.4 lista os estilos de páginas pré-definidos.
É possível alterar o estilo da página actual com o comando

`\thispagestyle{`*estilo*`}`

Uma descrição sobre como criar os seus próprios cabeçalhos e rodapés pode ser encontrado no *The LaTeX Companion* [3] e na secção 4.4 da página 72.

1.8 Grandes Projectos

Quando trabalha com documentos grandes, possivelmente gostava de separar os ficheiros de código em várias partes. O LaTeX tem dois comandos que ajudam a atingir este objectivo.

`\include{`*nomedoficheiro*`}`

pode usar este comando no corpo do seu documento para incluir o conteúdo de outro ficheiro chamado *nomedoficheiro.tex*. Note que o LaTeX iniciará uma nova página antes de processar o material deste ficheiro.

O segundo comando pode ser utilizado no preâmbulo. Ele permite instruir o LaTeX para incluir apenas alguns dos ficheiro incluídos.

`\includeonly{`*nomedoficheiro*`,`*nomedoficheiro*`,`...`}`

Depois deste comando ser executado no preâmbulo do documento, apenas os comandos `\include` para os ficheiros listados no argumento deste comando serão executados. Note que não podem existir espaços entre os nomes dos ficheiros e as vírgulas.

O comando `\include` inicia o texto, incluído numa nova página. Isto é útil quando usa o `\includeonly`, porque as mudanças de página não se moverão, mesmo quando alguns `\include`'s são omitidos. Algumas vezes isto pode não ser desejável. Neste caso, deve utilizar o comando

`\input{`*nomedoficheiro*`}`

que incluí simplesmente o ficheiro especificado, sem fatos brilhantes, nem cordas agarradas!

Para fazer o LaTeX verificar rapidamente o seu documento pode utilizar o pacote syntonly que faz com que o LaTeX percorra o ficheiro verificando

Tabela 1.4: Os Estilos de Páginas Pré-definidos no LaTeX.

`plain` imprime o número da página no fundo da página, no centro do rodapé. Este é o estilo por omissão.

`headings` imprime o nome do capítulo actual e o número da página no cabeçalho de cada página, enquanto que o rodapé se mantém vazio. (Este é o estilo usado neste documento)

`empty` coloca quer o cabeçalho quer o rodapé vazios.

apenas a sintaxe e utilização de comandos, mas não produz nenhum ficheiro. Como o LaTeX corre mais depressa neste modo, pode poupar-lhe muito do seu precioso tempo. A sua utilização é muito simples:

```
\usepackage{syntonly}
\syntaxonly
```

Quando precisar de produzir páginas, adicione um comentário à segunda linha (adicionando um símbolo de percentagem).

Capítulo 2

Escrever Texto

Depois de ler o capítulo anterior, deve saber as partes básicas que constituem um documento LaTeX 2ε. Este capítulo preencherá o resto da estrutura que precisa de conhecer para produzir material autêntico.

2.1 A Estrutura de um Texto

By Hanspeter Schmid <hanspi@schmid-werren.ch>

Ao escrever um texto, o principal ponto (alguma literatura DAAC[1] moderna excluída), é apresentar ideias, informação ou conhecimento ao leitor. Este irá compreender melhor o texto se estas ideias estiverem bem estruturadas, e irá ver e sentir melhor esta estrutura se a forma tipográfica reflectir a estrutura lógica e semântica do conteúdo.

O LaTeX é diferente de todos os outros sistemas de escrita apenas por ter de indicar a estrutura lógica e semântica do texto. Utilizando esta estrutura, deriva a forma tipográfica do texto de acordo com as "regras" dadas na classe do documento e nos vários ficheiros de estilo.

A unidade mais importante de texto em LaTeX (e em tipografia) é o parágrafo. Chamamos-lhe "unidade de texto" porque um parágrafo é a forma tipográfica que deve reflectir um pensamento coerente, ou uma ideia. Nas secções seguintes aprenderá como forçar mudanças de linha, por exemplo com \\ e mudanças de parágrafo deixando uma linha em branco no código. Se começa a descrever um novo pensamento, deve iniciar um novo parágrafo; se não, apenas mudanças de linha devem ser utilizadas. Em dúvida se deve ou não realizar quebras de parágrafos, pense no seu texto como um armário de ideias e pensamentos. Se tem uma quebra de parágrafo, mas o pensamento anterior continua, deve ser removido. Se alguma linha de pensamento completamente nova ocorre no mesmo parágrafo, então deve ser quebrado.

[1]Different At All Cost (diferente a todo o custo), uma tradução da UVA Suíça-Alemã (Um's Verrecken Anders).

A maior parte das pessoas esquece completamente a importância de quebras de parágrafos bem colocadas. Muitas pessoas nem sequer saber o significado duma quebra de parágrafo, e, especialmente em LaTeX, introduzem parágrafos sem saber. Este erro é especialmente fácil de fazer se forem utilizadas equações no texto. Veja os exemplos seguintes, e descubra porque algumas linhas em branco (quebras de parágrafos) são utilizados antes e depois da equação, e outras vezes não. (Se ainda não compreender todos os comandos o suficiente para compreender estes exemplos, continue a ler este e os capítulos seguintes, e depois volte a ler esta secção.)

```
% Example 1
\ldots quando Einstein introduziu a sua fórmula
\begin{equation}
  e = m \cdot c^2 \; ,
\end{equation}
que é ao mesmo tempo a mais conhecida e a menos
compreendida fórmula física.


% Example 2
\ldots de onde segue a lei actual de Kirchhoff:
\begin{equation}
  \sum_{k=1}^{n} I_k = 0 \; .
\end{equation}

A lei de voltagem de Kirchhoff pode ser deduzida\ldots


% Example 3
\ldots que tem várias vantagens.

\begin{equation}
  I_D = I_F - I_R
\end{equation}
é o centro de um modelo diferente de transistor.\ldots
```

A unidade mais pequena de texto seguinte é a frase. Em textos ingleses[2], existe um maior espaço depois de um ponto que acaba uma frase do que um que acaba uma abreviatura. O LaTeX tenta adivinhar qual dos casos deve utilizar. Se o LaTeX errar, deve dizer-lhe o que quer. Como fazer isto é explicado mais tarde neste capítulo.

A estruturação do texto extende-se até às partes de uma frase. Quase todas as línguas têm regras de pontuação muito complicadas, mas na sua

[2] NT: e também em portugueses

maioria (incluindo alemão e inglês[3]), irá obter quase todas as vírgulas correctas se se lembrar o que ela representa: uma pequena paragem no fluxo da língua. Se não tem a certeza onde colocar a vírgula, leia a frase em voz alta, e faça uma pequena respiração em cada vírgula. Se fica mal em algum sítio, apague a vírgula; se precisa de respirar (ou fazer uma pequena pausa) em algum outro ponto, insira uma vírgula.

Finalmente, os parágrafos de um texto devem também estar estruturados logicamente a um nível superior, encaixando-os em capítulos, secções, subsecções, e assim sucessivamente. No entanto, o efeito tipográfico de escrever, por exemplo, `\section{A Estrutura e Linguagem do Texto}` é tão obvio que é evidente quando estas estruturas de alto nível devem ser utilizadas.

2.2 Quebras de Linha e Quebras de Página

2.2.1 Parágrafos Justificados

É habitual que os livros sejam escritos de forma a que cada linha tenha o mesmo comprimento. O LaTeX insere as quebras de linha e espaços necessários entre palavras optimizando os conteúdos de um parágrafo, como um todo. Se necessário, ele também hifeniza as palavras que não cabem confortavelmente numa linha. Como os parágrafos são impressos depende na classe do documento. Normalmente a primeira linha do parágrafo é indentada, e não existe espaço adicional entre dois parágrafos. Veja a secção 5.3.2 para mais informação.

Em casos especiais pode ser necessário ordenar ao LaTeX para quebrar a linha:

`\\` ou `\newline`

iniciam uma nova linha sem iniciar um novo parágrafo.

`\\*`

proíbe, adicionalmente, uma quebra de página após a quebra de linha forçada.

`\newpage`

inicia uma nova página.

`\linebreak[`*n*`]`, `\nolinebreak[`*n*`]`, `\pagebreak[`*n*`]` e `\nopagebreak[`*n*`]`

fazem o que o seu nome diz. Permitem que o autor influencie as suas acções

[3] NT: e em português

com um argumento opcional *n* que pode variar desde zero até quatro. Ao colocar *n* com um valor inferior a 4 permite que o LaTeX ignore o seu comando se o resultado ficar muito feio. Não confunda estes comandos “break” com os comandos “new”. Mesmo ao indicar um comando “break”, o LaTeX tenta ainda preencher as bordas do texto e o comprimento total da página, como descrito na secção seguinte. Se realmente quer começar uma “nova linha”, então utilize o comando correspondente. Adivinhe qual!

O LaTeX tenta sempre produzir as melhores quebras de linha possíveis. Se não encontra uma forma de quebrar a linha dum modo que esteja de acordo com a sua qualidade, deixa uma linha sair um pouco da margem direita do parágrafo. No entanto, o LaTeX queixa-se (“overfull hbox”) ao processar o documento. Isto acontece mais frequentemente quando o LaTeX não encontra um sítio correcto para hifenizar a palavra.[4] Existe a possibilidade de instruir o LaTeX a baixar a sua qualidade um pouco dando-lhe o comando `\sloppy`, que previne estas linhas demasiado compridas aumentando o espaço entre palavras—mesmo que o resultado final não seja óptimo. Neste caso, um aviso (“underfull hbox”) é dado. Na maior parte destes casos o resultado não é o melhor. O comando `\fussy` traz o LaTeX de volta ao seu comportamento habitual.

2.2.2 Hifenização

O LaTeX hifeniza as palavras sempre que precisa. Se o algoritmo de hifenização não encontrar o ponto correcto, pode remediar a situação utilizando o seguinte comando, para indicar ao TeX a excepção.

O comando

`\hyphenation{`*lista de palavras*`}`

causa as palavras listadas no argumento sejam hifenizadas apenas nos pontos marcados por “`-`”. O argumento do comando deve conter apenas palavras constituídas de letras normais ou sinais considerados como letras normais pelo LaTeX. As sugestões de hifenização são guardados para a língua activa quando o comando de hifenização ocorre. Isto significa que se colocar um comando de hifenização no preâmbulo do seu documento, influenciará a hifenização da língua inglesa. Se colocar o comando após o `\begin{document}` e está a usar algum pacote para suporte de outras línguas como seja o pacote babel, então as sugestões de hifenização estarão activas na língua activada pelo babel.

[4] Apesar do LaTeX avisá-lo quando isto acontece (Overfull hbox) e mostrar a linha problemática, estas linhas nem sempre são fáceis de encontrar. Se utilizar a opção `draft` no comando `\documentclass`, estas linhas serão marcadas com uma linha preta na margem direita.

O seguinte exemplo permitirá que "hifenização" seja hifenizado tão bem como "Hifenização", e previne de todo que "FORTRAN", "Fortran" e "fortran" sejam hifenizados. No argumento deste comando não são permitidos símbolos ou caracteres especiais.

Exemplo:

```
\hyphenation{FORTRAN Hi-fe-ni-za-ção}
```

O comando `\-` insere um hífen descritivo numa palavra. Estes passam também a ser os únicos pontos de hifenização permitidos na palavra. Este comando é especialmente útil para palavras que contém caracteres especiais (por exemplo, caracteres acentuados), porque o LaTeX não hifeniza automaticamente palavras que contenham esses caracteres.

```
Penso que isto é: su\-per\-cal\-%
i\-frag\-i\-lis\-tic\-ex\-pi\-%
al\-i\-do\-cious
```

Penso que isto é: supercalifragilisticexpiali-
docious

Várias palavras podem ser mantidas juntas numa linha, utilizando o comando

`\mbox{`*texto*`}`

que obriga o LaTeX a manter o seu argumento junto em todas as circunstâncias.

```
O meu número de telefone irá mudar
brevemente para \mbox{0116 291 2319}.

O parâmetro
\mbox{\emph{nomedoficheiro}} deverá
conter o nome do ficheiro.
```

O meu número de telefone irá mudar brevemente para 0116 291 2319.

O parâmetro *nomedoficheiro* deverá conter o nome do ficheiro.

O comando `\fbox` é idêntico ao `\mbox`, mas adiciona um rectângulo desenhado à volta do conteúdo.

2.3 Texto já Feito

Em alguns dos exemplos nas páginas anteriores deverá ter visto alguns comandos LaTeX muito simples para escrever texto especial:

Comando	Exemplo	Descrição
`\today`	12 de Dezembro de 2002	Data actual na língua actual
`\TeX`	TeX	O nome do seu tipógrafo preferido
`\LaTeX`	LaTeX	O Nome do Jogo
`\LaTeXe`	LaTeX 2_ε	A incarnação actual do LaTeX

2.4 Símbolos e Caracteres Especiais

2.4.1 Aspas

Não deve utilizar o caracter `"` para desenhar aspas como o faria numa máquina de escrever. Existem duas marcas especiais para abrir e fechar aspas em tipografia. Em LaTeX, use dois ` (acento agudo) para abrir aspas e dois ' (apóstrofe) para fechar aspas. Para aspas simples use apenas um de cada.

```
``Por favor, pressione a tecla `x'.''
```

"Por favor, pressione a tecla 'x'."

2.4.2 Traços e Hífens

O LaTeX conhece quatro tipos diferentes de traços. Pode aceder três deles colocando um, dois ou três `-` consecutivos. O quarto sinal não é um traço—é o sinal matemático menos:

```
fim-de-semana, segunda-feira\\
páginas 13--67\\
sim---ou não? \\
$0$, $1$ e $-1$
```

fim-de-semana, segunda-feira
páginas 13–67
sim—ou não?
0, 1 e -1

Os nomes destes traços são: '-' hífen, '–' en-dash, '—' em-dash e '$-$' sinal de menos.

2.4.3 Til ($\sim$)

Um caracter visto frequentemente em endereços de Internet é o til. Para gerar este símbolo em LaTeX pode usar `\~` mas o resultado: ˜ não é propriamente o que queria. Tente desta forma:

```
http://www.rich.edu/\~{}bush \\
http://www.clever.edu/$\sim$demo
```

http://www.rich.edu/˜bush
http://www.clever.edu/$\sim$demo

2.4.4 Símbolo de Graus ($\circ$)

O seguinte exemplo mostra como imprimir um símbolo de graus em LaTeX:

```
Temperatura: $-30\,^{\circ}\mathrm{C}$,
Daqui a pouco estou a congelar.
```

Temperatura: $-30\,^{\circ}\mathrm{C}$, Daqui a pouco estou a congelar.

2.4.5 O Símbolo de Euro (€)

Ultimamente, ao escrever sobre dinheiro vai precisar do símbolo do euro. Muitos dos tipos de letra actuais contêm o símbolo do Euro. Depois de carregar o pacote textcomp no preâmbulo do seu documento

```
\usepackage{textcomp}
```

pode usar o comando

```
\texteuro
```

para o desenhar.

Se o seu tipo de letra não providenciar o seu próprio símbolo do Euro ou se não gosta do símbolo do Euro nos tipos de letra que possuí, ainda tem duas hipóteses:

Em primeiro lugar, o pacote eurosym que contém o símbolo oficial do Euro:

```
\usepackage[official]{eurosym}
```

Se preferir um símbolo mais parecido com o seu tipo de letra, use a opção `gen` no lugar da opção `official`.

O pacote marvosym também incluí muitos símbolos diferentes, incluindo o Euro, sob o nome:

```
\EUR
```

```
The \texteuro{}, the \euro{}
and the \EUR{} all look
different.
```

The €, the € and the € all look different.

2.4.6 Reticencias (. . .)

Numa máquina de escrever, uma vírgula ou um ponto ocupa o mesmo espaço de qualquer outra letra. Ao imprimir livros, estes caracteres ocupam apenas um pequeno espaço e são colocados muito próximos à letra precedente. Desta forma, não pode introduzir 'reticencias' simplesmente introduzindo três pontos, porque o espaçamento estará errado. No entanto, existe um comando especial para estes pontos. É chamado

```
\ldots
```

```
Não desta forma ... mas assim:\\
Nova Iorque, Tóquio, Budapeste, \ldots
```

Não desta forma ... mas assim:
Nova Iorque, Tóquio, Budapeste, ...

2.4.7 Ligações

Algumas combinações de letras são escritas, não colocando as diferentes letras uma após a outra, mas usando símbolos especiais.

ﬀ ﬁ ﬂ ﬃ... em vez de ff fi fl ffi ...

Estas chamadas 'ligações' podem ser proibidas inserindo uma `\mbox{}` entre as duas letras em questão. Isto pode ser necessário com palavras construídas de duas palavras.

```
\Large Not shelfful\\
but shelf\mbox{}ful
```

Not shelﬀul
but shelfful

2.4.8 Caracteres Especiais e Acentos

O LaTeX suporta o uso de acentos e caracteres especiais de muitas línguas. A tabela 2.1 mostra todos os tipos de acentos sendo aplicados à letra o. Naturalmente que outras letras também funcionam.

Para colocar um acento no topo de um i ou de um j, o ponto deve ser removido. Isto é conseguido escrevendo `\i` e `\j`.

```
H\^otel, na\"\i ve, \'el\`eve,\\
sm\o rrebr\o d, !`Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

2.5 Suporte de Línguas Internacionais

Se precisa de escrever documentos noutras línguas que não o inglês, então existem três áreas onde o LaTeX pode ser configurado apropriadamente:

1. Todos os textos gerados automaticamente[5] podem ser adaptados para a outra língua. Para a maior parte das línguas, estas mudanças podem ser conseguidas utilizando o pacote babel por Johannes Braams.

[5]Tabela de Conteúdos, Lista de Figuras, ...

2. O LaTeX precisa de saber as regras de hifenização para a nova língua. Conseguir as regras de hifenização no LaTeX é um pouco mais complicado. Obriga a reconstruir o ficheiro de formato activando padrões de hifenização diferentes. O seu *Local Guide* [4] deve apresentar mais informação na realização desta tarefa.

3. Regras tipográficas específicas de algumas línguas. Em Francês, por exemplo, existe um espaço obrigatório antes de cada caracter de dois pontos (:).

Se o seu sistema já está configurado de forma apropriada, pode activar o pacote babel adicionando o comando

`\usepackage[`*língua*`]{babel}`

depois do comando `\documentclass`. A lista das *língua*s que o seu sistema suporta serão mostradas sempre que o seu compilador inicia. O Babel irá activar automaticamente as regras apropriadas de hifenização para a língua de escolher. Se o seu formato LaTeX não suporta hifenização na língua que escolheu, o babel continuará a funcionar mas irá desactivar a hifenização o que terá um efeito bastante negativo na aparencia visual do resultado do documento.

Tabela 2.1: Caracteres especiais e Acentos.

ò	`\‘o`	ó	`\’o`	ô	`\^o`	õ	`\~o`
ō	`\=o`	ȯ	`\.o`	ö	`\"o`	ç	`\c c`
ŏ	`\u o`	ǒ	`\v o`	ő	`\H o`	ǫ	`\c o`
ọ	`\d o`	o̲	`\b o`	o͡o	`\t oo`		
œ	`\oe`	Œ	`\OE`	æ	`\ae`	Æ	`\AE`
å	`\aa`	Å	`\AA`				
ø	`\o`	Ø	`\O`	ł	`\l`	Ł	`\L`
ı	`\i`	ȷ	`\j`	¡	`!‘`	¿	`?‘`

Se inicializar o babel com várias línguas

`\usepackage[`*línguaA*`,`*línguaB*`]{babel}`

irá precisar de usar o comando

`\selectlanguage{`*línguaA*`}`

para definir a língua actual.

Para algumas línguas, o babel também especifica novos comandos que simplifica a introdução de caracteres especiais. O alemão, por exemplo, contém letras com *umlautes* (äöü). Com o babel, pode introduzir um ö escrevendo `"o` em vez de `\"o`.

Quase todos os sistemas modernos permitem introduzir alguns caracteres especiais directamente pelo teclado. O LaTeX consegue utilizar estes caracteres. Desde que o LaTeX 2$_\varepsilon$ foi disponibilizado em Dezembro de 1994, o suporte para vários tipos de codificação está incluído na distribuição básica do LaTeX 2$_\varepsilon$. Verifique o pacote inputenc:

`\usepackage[`*codificação*`]{inputenc}`

Ao utilizar este pacote, deverá considerar que outras pessoas podem não conseguir ler o código do seu documento no seu computador, porque utilizam uma codificação diferente. Por exemplo, o umlaut alemão ä em OS/2 é codificado como 132, mas em alguns sistemas Unix utilizando ISO-LATIN 1 é codificado como 228. Desta forma deve utilizar esta funcionalidade com cuidado. As seguintes codificações podem ser úteis, dependendo do tipo de sistema com que está a trabalhar: Mac – `applemax`; Unix – `latin1`; Windows – `ansinew`.

Sistema Operativo	Codificação
Mac	`applemac`
Unix	`latin1`
Windows	`ansinew`
DOS, OS/2	`cp850`

A codificação de tipos de letra é uma matéria diferente. Esta, define em que posições cada letra é guardada dentro da TeX-font. O tipo de letra original Computer Modern TeXapenas contém os 128 caracteres do antigo conjunto de caracteres ASCII de 7-bit. Quando são necessários caracteres acentuados, o TeX cria-os combinando caracteres normais com os respectivos acentos. Enquanto que o resultado pode parecer perfeito, esta abordagem pára a hifenização automática dentro de palavras que contém caracteres acentuados.

Felizmente, a maior parte das distribuições modernas do TEX contém uma cópia dos tipos de letra EC. Estas são idênticas às Computer Modern, mas contém caracteres especiais para a maior parte dos caracteres acentuados utilizados nas línguas europeias. Ao utilizar teste tipo de letra pode melhorar a hifenização de documentos não ingleses. Este tipo de letras pode ser activado incluindo o pacote fontenc no preâmbulo do seu documento.

```
\usepackage[T1]{fontenc}
```

2.5.1 Suporte para Alemão

Aqui vão algumas dicas para aqueles que estejam a criar documentos LATEX em Alemão: pode carregar o suporte para língua alemã com o comando

```
\usepackage[german]{babel}
```

Isto fará com que o LATEX passe a fazer hifenização para a língua Alemã, se tiver o seu sistema LATEX bem configurado. Também irá alterar todo o texto automático para alemão. Por exemplo, "Chapter" irá passar a "Kapitel." Um conjunto de novos comandos irão passar a estar disponíveis, que lhe irão permitir introduzir texto alemão de forma mais rápida, mesmo que não use o pacote inputenc. Repare na tabela 2.2 para alguma inspiração. Com o inputenc, tudo isto se torna desnecessário, mas o seu texto também irá ficar preso num mundo de codificação particular.

Tabela 2.2: Caracteres especiais alemães.

`"a`	ä	`"s`	ß
`"‘`	„	`"’`	“
`"<` or `\flqq`	«	`">` or `\frqq`	»
`\flq`	■	`\frq`	■
`\dq`	”		

Em livros alemães irá encontrar aspas de estilo francês («guillemets»). Os tipógrafos alemães usam-as de forma diferente. Uma frase entre aspas num livro alemão deverá ser »assim «. Na parte da Suíça que fala alemão, os tipógrafos usam «guillemets» da mesma forma que os Franceses.

Um grande problema aparece do uso de comandos como `\flq`: se usar um tipo de letra OT1 (que é o tipo de letra por omissão) as guillemets irão parecer o símbolo matemático "$\ll$", que irá revoltar um estômago de

tipógrafo. Os tipos de letra codificados em T1, por outro lado, não contêm os símbolos necessários. Assim, se está a usar este tipo de aspas, tenha a certeza que está com a codificação T1. (`\usepackage[T1]{fontenc}`)

2.5.2 Suporte para Português

By Alberto Simões <albie@alfarrabio.di.uminho.pt>

Caracteres Acentuados

Já se torna fácil utilizar caracteres *latin-1* directamente no código fonte do seu documento. Para saber como configurar o teclado para permitir inserir estes caracteres em Linux, visite `http://gil.di.uminho.pt`. Para que o LaTeX não se queixe, active o pacote fontenc da seguinte forma:

```
\usepackage[T1]{fontenc}
```

Depois, usar o LaTeX como está habituado!

Hifenização

Nas distribuições do LaTeX em Linux utilizadas até agora, nunca encontrei uma com a hifenização automática para português activada quando se usa o pacote babel. No entanto, a sua activação é extremamente simples se tem acesso à aplicação `texconfig`. Tente executá-la (como administrador do sistema) e, caso não a encontre, verifique se não estará escondida algures, como em `/usr/share/texmf/bin/texconfig`.

Depois de executar a aplicação, e após alguns segundos de processamento, aparecerá uma janela com várias opções. Escolha a opção de hifenização (`HYPHEN` - hyphenation). Na nova janela, escolha a opção `latex` que abrirá o editor `vi` com um ficheiro de configuração.

Agora deve procurar a palavra `portuges` neste ficheiro. Se não está habituado a usar este editor, tecle `/` seguido de `portuges`. O cursor aparecerá sobre uma linha da forma:

```
%portuges        pt8hyph.tex
```

Repare no símbolo de comentário (símbolo de percentagem). Se existe, significa que a hifenização portuguesa está desligada. Se seguiu as instrucções dadas, tecle `x`. Caso contrário, não precisa de alterar nada. Esta linha deve passar a:

```
portuges        pt8hyph.tex
```

Para terminar, escreva `:wq`. O sistema começará a processar este ficheiro. Espere até que volte a aparecer a janela inicial e escolha a opção para sair (`Exit`).

2.5.3 Suporte para Francês

By Daniel Flipo <daniel.flipo@univ-lille1.fr>

Alguns conselhos para aqueles que desejem criar documentos em Francês usando o LaTeX. Pode carregar o suporte para o Francês usando o comando:

```
\usepackage[frenchb]{babel}
```

Note que, por razões histórias, o nome da opção do pacote babel para o Francês é *frenchb* ou *francais* mas nunca *french*.

Esta opção inicia a hifenização Francesa, se tiver o seu LaTeX configurado de forma correcta. Também altera todo o texto automático para francês: `\chapter` escreve "Chapitre", `\today` escreve a data actual em francês, e assim sucessivamente. Um grande conjunto de novos comandos passam a existir, que lhe permitem escrever ficheiros em Francês de forma mais simples. Dê uma olhadela à tabela 2.3 para inspiração.

Tabela 2.3: Comandos especiais para Francês.

`\og guillemets \fg{}`	« guillemets »
`M\up{me}, D\up{r}`	Mme, D^{r}
`1\ier{}, 1\iere{}, 1\ieres{}`	1er, 1re, 1res
`2\ieme{} 4\iemes{}`	2e 4^{es}
`\No 1, \no 2`	N° 1, n° 2
`20~\degres C, 45\degres`	20 °C, 45°
`\bsc{M. Durand}`	M. DURAND
`\nombre{1234,56789}`	1 234,567 89

Também irá reparar que o formato das listas mudam quando usa a língua francesa. Para ficar a conhecer tudo o que a opção `franchb` do pacote babel faz e como pode alterar o seu funcionamento, execute o LaTeX no ficheiro `frenchb.dtx` e leia o ficheiro produzido: `frenchb.dvi`.

2.5.4 Suporte para Coreano[6]

Para usar o LaTeX para escrever Coreano, precisamos de resolver três problemas:

[6]Considerando o número de questões que os utilizadores de LaTeX Coreanos têm de ter em conta, esta secção foi escrita por Karnes KIM da equipa de gradução do lshort para Coreano. Foi traduzida para Inglês por SHIN Jungshik e encolhida por Tobi Oetiker

1. Temos de ser capazes de editar ficheiros LaTeX em coreano. Estes ficheiros devem ser em formato de texto puro, mas porque o Coreano usa os seus próprios caracteres fora do reportório do US-ASCII, eles irão parecer bastante estranhos num editor ASCII comum. As duas codificações mais usadas para textos em Coreano são EUC-KR e a sua extensão compatível usada no MS-WIndows Coreano, CP949/Windows-949/UHC. Nestas codificações, cada caracter US-ASCII representa o seu caracter ASCII normal semelhante a outras codificações compatíveis como a ISO-8859-x, EUC-JP,Shift_JIS, e Big5. Por outro lado, sílabas Hangul, Hanjas (caracteres Chineses usados na Coreia), Hangul Jamos, Hirakanas, Katakanas, Grego e caracteres Cirílicos e outros símbolos e letras desenhadas pelo KS X 1001 são representados por dois octetos consecutivos. O primeiro tem o seu conjunto MSB. Até ao meio dos 1990's, demorou uma quantidade de tempo e trabalho considerável para criar um ambiente Coreano num sistema operativo não Coreano. Pode dar uma olhadela ao agora muito antigo `http://jshin.net/faq` para ter uma ideia de como se usava Coreano num sistema operativo não Coreano no meio dos 1990's. Nestes dias, os três principais sistemas operativos (Mac OS, Unix, Windows) vêem equipados com suporte multilingue bastante decente, e internacionalização de forma que editar um texto em Coreano já não é tanto um problema, mesmo em sistemas operativos não coreanos.

2. O TeX e o LaTeX foram originalmente escritos para línguas com menos de 256 caracteres no seu alfabeto. Para os fazer funcionar para línguas com consideravelmente mais caracteres como o Coreano [7] ou Chinês, um mecanismo de sub-tipos de letra foi desenvolvido. Divide uma letra

[7] O Hangul Coreano é uma língua alfabética com 14 consoantes básicas e 10 vogais básicas (Jamos). De forma diferente das línguas latinas ou cirílicas, cada caracter individual tem de ser colocado em caixas rectangulares com um tamanho similar ao dos caracteres Chineses. Cada caixa representa uma sílaba. Um número ilimitado de sílabas pode ser formado deste conjunto finito de vogais e consoantes. Os standards modernos da ortografia Coreana (na Coreia do Sul e na Coreia do Norte), colocam algumas restrições na formação destas caixas. Assim, apenas um conjunto finito de silabas ortograficamente correctas existem. A codificação de caracteres coreanos define um código individual para cada uma destas sílabas (KS X 1001:1998 e KS X 1002:1992). O Hangul passa a ser tratado como nos sistemas de escrita do Chinês ou do Japonês, com centenas de caracteres ideográficos. O ISO 10646/Unicode oferece ambas as formas de representar Hangul usado para o Coreano *moderno* codificando Conjoining Hangul Jamos (alfabetos: `http://www.unicode.org/charts/PDF/U1100.pdf`) em adição à total codificação das sílibas ortograficamente permitidas em Coreano *moderno* (`http://www.unicode.org/charts/PDF/UAC00.pdf`). Um dos maiores desafios na tipografia Coreana com LaTeX e sistemas similares é o suporte para Coreano mais antigo—e possívelmente Coreano do futuro—onde as sílabas só podem ser representadas juntando Jamos em Unicode. Espera-se que futuros motores do TeX como Ω e Λ irão, eventualmente, providenciar soluções de forma a que alguns linguistas e historidores Coreanos irão desistir do MS Word que já tem um bom suporte para Coreano antigo.

CJK com centenas ou dezenas de centenas de glifos num conjunto de sub-tipos com 256 glifos cada. Para o Coreano, existem três pacotes bastante usados; HLaTeX por UN Koaunghi, hLaTeXp por CHA Jae-choon e o pacote CJK por Werner Lemberg.[8] HLaTeX e hLaTeXp são específicos para o Coreano e disponibilizam o Coreano sobre o suporte de tipos de letras. Ambos conseguem processar texto Coreano codificado em EUC-KR. HLaTeX pode até processar ficheiros codificados em CP949/Windows-949/UHC e UTF-8 quando usado com Λ ou Ω.

O pacote CJK não é específico para Coreano. Consegue processar ficheiros em UTF-8 assim como em várias codificações CJK incluindo EUC-KR e CP949/Windows-949/UHC. Pode ser usado para escrever documentos com conteúdo multi-lingue (especialmente Chinês, Japonês e Coreano). O Pacote CJK não tem localização em Coreano como a oferecida pelo HLaTeX e não disponibiliza tantos tipos de letra Coreana como o HLaTeX.

3. O último objectivo de usar programas como o TeX e o LaTeX é para obter documentos "esteticamente" satisfatórios. Sem dúvida que o elemento mais importante é o tipo de letra bem desenhado. A distribuição HLaTeX inclui tipos de letra UHC PostScript de 10 famílias diferentes e tipos Munhwabu[9] de 5 famílias diferentes. O pacote CJK funciona com um conjunto de tipos de versões anteriores ao HLaTeX e também consegue usar tipos de letra Bitstream cyberbit TrueType.

Para usar o pacote HLaTeX para escrever o seu texto em Coreano, inclua a seguinte declaração no preâmbulo do documento:

```
\usepackage{hangul}
```

Este comando liga a localização para Coreano. Os cabeçalhos de capítulos, secções, secções, índices de conteúdos ou figuras, todos eles irão ser traduzidos para Coreano, e a formatação do documento irá seguir as convenções Coreanas. O pacote também contempla uma "selecção de partículas" automática. Em Coreano, existem pares de partículas pós-fixas gramaticamente equivalentes mas diferentes em forma. Qual é correcto depende se a sílaba precedente acaba em vogal ou consoante. (É um pouco mais complexo que isto, mas deve estar a ter uma boa ideia.) Coreanos nativos não têm problemas a escolher a partícula correcta, mas o TeX não pode determinar que particula usar para referências ou outro texto automático que irá mudar ao editar o documento. Esta selecção de partículas manual gera algum

[8] Podem ser obtidos em `CTAN:/tex-archive/language/korean/HLaTeX/` `CTAN:/tex-archive/language/korean/CJK/` e `http://knot.kaist.ac.kr/htex/`

[9] Ministério da Cultura Coreano

esforço em colocar as partículas cada vez que adiciona ou remove referências ou simplesmente baralha alguma das partes do seu documento. HLaTeX alivia os seus utilizadores deste processo aborrecido e delicado.

No caso de não precisar das características de localização do Coreano, e quer apenas escrever em Coreano, pode usar a seguinte linha no preâmbulo do seu documento,

```
\usepackage{hfont}
```

Para maior detalhe sobre o uso de Coreano com HLaTeX, consultar o *HLaTeX Guide*. Veja também a página do grupo Coreano de utilizadores de TeX (KTUG) em `http://www.ktug.or.kr/`. Também existe uma tradução Coreana deste manual.

2.6 O Espaço entre Palavras

Para obter no resultado uma margem direita alinhada, o LaTeX insere várias quantidades de espaço entre palavras. Insere um pouco mais de espaço no fim das frases, visto que faz o texto ficar mais legível. O LaTeX assume que as frases acabam com pontos finais, de interrogação ou de exclamação. Se o ponto final seguir uma letra maiúscula, esta não é tomada como um fim de frase, visto que pontos após letras maiúsculas ocorrem, normalmente, em abreviaturas.

Alguma excepção a estas regras deve ser especificada pelo autor. Um *backslash* antes de um espaço gera um espaço que não será alargado. Um til '~' gera um espaço que não pode ser alargado e que, adicionalmente, proíbe uma mudança de linha na respectiva posição. O comando `\@` antes de um ponto especifica que esse ponto termina uma frase, mesmo que se lhe siga uma letra maiúscula.

```
O Sr~Smith estava contente de a ver\\
conforma a fig.~5\\
Eu gosto de BASIC\@. E tu?
```

O Sr Smith estava contente de a ver
conforma a fig. 5
Eu gosto de BASIC. E tu?

O espaço adicional após pontos pode ser desactivado com o comando

```
\frenchspacing
```

que diz ao LaTeX para *não* inserir mais espaço depois de ponto do que em relação a qualquer outro caracter. Isto é muito comum em línguas não inglesas, excepto em bibliografias. Se usar `\frenchspacing`, o comando `\@` não é necessário.

2.7 Títulos, Capítulos e Secções

Para ajudar o leitor a encontrar a linha de leitura ao longo do documento, deve dividi-lo em capítulos, secções e subsecções. O LaTeX permite que se faça isto com comandos especiais que tomam o título como seu argumento. Agora, é consigo que os use na ordem correcta.

Os comandos de divisão do texto que estão disponíveis para a class `article` são:

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

Quando precisa de dividir o seu documento em partes sem influenciar a numeração de secções ou capítulos pode usar

```
\part{...}
```

Se estiver a trabalhar com as classes `report` ou `book`, um comando adicional para seccionar ao nível de topo, torna-se disponível

```
\chapter{...}
```

Pode utilizar um comando extra para dividir o seu texto para a classe `report` ou `book`: `\chapter{...}`

Se desejar separar os seus documentos em partes sem influenciar a numeração de capítulo/secção pode usar `\part{...}`.

Como a classe `article` não sabe nada acerca de capítulos, torna-se muito fácil adicionar artigos como capítulos num livro. O espaçamento entre secções, a numeração e o tamanho de letra dos títulos serão colocados automáticamente pelo LaTeX.

Dois destes comandos são ligeiramente especiais:

- O comando `\part` não influencia a numeração de sequência dos capítulos;

- O comando `\appendix` não leva nenhum argumento. Apenas muda a numeração de capítulos para letras.[10]

O LaTeX cria um tabela de conteúdos pegando nos títulos de secção e no número de página do último cíclo de compilação do documento. O comando

[10]Para o estilo `article`, muda a numeração de secções.

`\tableofcontents`

expande-se para uma tabela de conteúdos no sítio onde for invocado. Um novo documento deve ser compilado ("LaTeXado") duas vezes para obter uma tabela de conteúdos correcta. Algumas vezes, pode ser necessário compilar o documento uma terceira vez. O LaTeX avisará quando isto for necessário.

Todos os comandos listados acima também existem em versões "estreladas". Uma versão "estrelada" do comando é construída adicionando uma estrela `*` após o nome do comando. Estas versões geram títulos que não aparecerão na tabela de conteúdos e que não serão numerados. O comando `\section{Ajuda}`, por exemplo, pode passar a `\section*{Ajuda}`.

Normalmente, o título da secção aparecerá na tabela de conteúdos exactamente como introduziu no texto. Algumas vezes isto não é possível por o título ser demasiado grande e a tabela de conteúdos não ficar legível. Então, a entrada que aparecerá na tabela de conteúdos pode ser especificada como um argumento opcional antes do verdadeiro título.

```
\chapter[Título pequeno para a tabela de conteúdos]{Um
  título grande e especialmente aborrecido, que aparece
  na página propriamente dita.}
```

O título do documento como um todo é gerado invocando o comando

`\maketitle`

Os conteúdos do título têm de ser definidos pelos comandos

`\title{...}`, `\author{...}` e opcionalmente `\date{...}`

antes de chamar o `\maketitle`. No argumento de `\author`, pode escrever vários nomes separados pelo comando `\and`.

Um exemplo de alguns dos comandos mencionados acima podem ser encontrados na figura 1.2 da página 8.

Além destes comandos, o LaTeX 2_ε introduziu três comandos adicionais para serem utilizados na classe `book`. São úteis para dividir a publicação. Estes, alteram os cabeçalhos de capítulo e numeração de página para funcionar como esperamos que o faça num livro:

`\frontmatter` deve ser o primeiro comando após `\begin{document}`. Irá mudar a numeração de página para numerais romanos. É normal usar os comandos de divisão do texto na sua forma "estrelada" para os capítulos que aparecem nesta parte do livro (exemplo `\chapter*{Prefácio}`) para fazer com que o LaTeX não os numere.

`\mainmatter` aparece exactamente antes do primeiro capítulo do livro. Muda a numeração para numerais árabes e coloca o contador de página a zero.

`\appendix` marca o inicio de material adicional no seu livro. Depois deste comando os capítulos serão numerados com letras.

`\backmatter` deve ser inserido antes dos últimos itens do seu livro como sejam a bibliografia e o índice. Nas classes de documento padrão, não tem qualquer efeito visual.

2.8 Referências Cruzadas

Em livros, relatórios e artigos, existem frequentemente referências cruzadas para figuras, tabelas e segmentos especiais de texto. O LaTeX providencia os seguintes comandos para realizar referências cruzadas:

`\label{`*marca*`}`, `\ref{`*marca*`}` e `\pageref{`*marca*`}`

onde *marca* é um identificador escolhido pelo utilizador. O LaTeX substituí `\ref` pelo número da secção, subsecção, figura, tabela ou teorema após o respectivo comando `\label` foi invocado. O `\pageref` imprime o número da página onde o comando `\label` ocorreu.[11] Tal como os títulos de secções, os números utilizados são os da compilação anterior.

```
Uma referência para esta subsecção
\label{sec:esta} aparece como:
``ver secção~\ref{sec:esta} na
página~\pageref{sec:esta}.''
```

Uma referência para esta subsecção aparece como: "ver secção 18 na página 35."

2.9 Notas de Rodapé

Com o comando

`\footnote{`*texto na nota de rodapé*`}`

é impressa uma nota de rodapé no fundo da página actual. Estas notas devem ser postas[12] após a palavra à qual a frase se refere. Notas de rodapé

[11] Note que estes comandos não sabem a que é que se referem. O `\label` apenas grava o último número gerado.

[12] "postas" pode ser uma forma do verbo pôr ou apenas bocados de peixe.

que se referem a frases ou partes delas, devem ser colocadas após a vírgula ou ponto. [13]

```
As notas de rodapé\footnote{Isto é
  uma nova de rodapé.} são muito
  usadas por utilizadores \LaTeX.
```

> As notas de rodapé[a] são muito usadas por utilizadores LaTeX.
>
> [a]Isto é uma nova de rodapé.

2.10 Palavras Realçadas

Se um texto é escrito utilizando uma máquina de escrever, `palavras importantes são salientadas` <u>`sublinhado-as`</u>.

`\underline{`*texto*`}`

Em livros impressos, no entanto, as palavras são realçadas escrevendo-as em *itálico*. O LaTeX tem o comando

`\emph{`*texto*`}`

que salienta o texto. O que o comando faz, na verdade, depende do seu contexto:

```
\emph{Se utilizar este
  comando dentro de um bocado
  de texto realçado, então o
  \LaTeX{} usa o tipo de letra
  \emph{normal} para salientar.}
```

> *Se utilizar este comando dentro de um bocado de texto realçado, então o LaTeX usa o tipo de letra* normal *para salientar.*

Note a diferença entre dizer ao LaTeX para *salientar* alguma coisa e dizer-lhe para usar um tipo de letra *diferente*:

```
\textit{Pode também
  \emph{salientar} texto se
  ele está em itálico,}
\textsf{num tipo de letra
  \emph{sans-serif},}
\texttt{ou num estilo
  \emph{máquina} de escrever.}
```

> *Pode também* salientar *texto se ele está em itálico,* num tipo de letra *sans-serif*, `ou num estilo` *`máquina`* `de escrever.`

[13]Note que as notas de rodapé são distracções para o leitor que o desviam do corpo principal do seu documento. Afinal, toda a gente lê as notas de rodapé, somos uma espécie curiosa. Portanto, porque não integrar tudo o que quer dizer no corpo do documento?[14]

[14]Uma carta não vai necessariamente para onde está endereçada :-).

2.11 Ambientes

```
\begin{ambiente}  texto  \end{ambiente}
```

Onde *ambiente* é o nome do ambiente. Os ambientes podem ser chamados várias vezes uns dentro dos outros desde que a ordem de chamada seja mantida.

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

Nas secções seguintes, todos os ambientes importantes serão explicados.

2.11.1 Indicar, Enumerar, e Descrever

O ambiente `itemize` é útil para listas simples, o `enumerate` para listas enumeradas e o `description` para descrições.

```
\flushleft
\begin{enumerate}
\item Pode misturar ambientes
de listas conforme o seu gosto:
\begin{itemize}
\item Mas pode começar a parecer
muito patético.
\item[-] Com um hífen,
\end{itemize}
\item Portanto, lembre-se: algo\ldots
\begin{description}
\item[Estúpido] não se transformará
em algo inteligente ao ser listado.
\item[Interessante] pode ser
apresentado lindamente numa lista.
\end{description}
\end{enumerate}
```

1. Pode misturar ambientes de listas conforme o seu gosto:
   - Mas pode começar a parecer muito patético.
   - \- Com um hífen,
2. Portanto, lembre-se: algo...

   Estúpido não se transformará em algo inteligente ao ser listado.

   Interessante pode ser apresentado lindamente numa lista.

2.11.2 Esquerda, Direita e Centro

Dois ambientes, `flushleft` e `flushright` geram parágrafos que estão alinhados à esquerda ou à direita. O ambiente `center` gera texto centrado. Se não colocar \\ para indicar as quebras de linha, o LaTeX irá determinar automaticamente onde elas devem ocorrer.

```
\begin{flushleft}
Este texto está\\ alinhado à esquerda.
\LaTeX{} não está a tentar fazer
cada linha do mesmo tamanho.
\end{flushleft}
```

Este texto está
alinhado à esquerda. LaTeX não está a tentar fazer cada linha do mesmo tamanho.

```
\begin{flushright}
Este é um texto alinhado\\
à direita. O \LaTeX{} não está a
tentar fazer cada linha do mesmo
 comprimento.
\end{flushright}
```

Este é um texto alinhado
à direita. O LaTeX não está a tentar fazer
cada linha do mesmo comprimento.

```
\begin{center}
No centro\\da terra
\end{center}
```

No centro
da terra

2.11.3 Citações e Versos

O ambiente `quote` é útil para citações, frases importantes e exemplos.

```
Uma regra tipográfica para
o comprimento de uma linha é:
\begin{quote}
Em média, nenhuma linha deverá
exceder 66~caracteres.

É por isto que as páginas \LaTeX{}
têm margens tão grandes.
\end{quote}
Por isso é que a impressão em
várias colunas é utilizada em
jornais.
```

Uma regra tipográfica para o comprimento de uma linha é:

> Em média, nenhuma linha deverá exceder 66 caracteres.
>
> É por isto que as páginas LaTeX têm margens tão grandes.

Por isso é que a impressão em várias colunas é utilizada em jornais.

Existem dois ambientes muito semelhantes: o `quotation` e o `verse`. O primeiro é útil para citações longas que são constituídas por vários parágrafos, porque os irá indentar. O ambiente `verse` é útil para poemas onde as mudanças de linha são importantes. As linhas são separadas enviando um \\ no fim de uma linha e uma linha em branco após cada verso.

```
Só conheço um poema em inglês.
É sobre Humpty Dumpty.
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\\
Humpty Dumpty had a great fall.\\
All the King's horses and all
the King's men\\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

Só conheço um poema em inglês. É sobre Humpty Dumpty.

> Humpty Dumpty sat on a wall:
Humpty Dumpty had a great fall.
All the King's horses and all the King's men
Couldn't put Humpty together again.

2.11.4 Tal & Qual

Os textos escritos entre `\begin{verbatim}` e `\end{verbatim}` serão passados directamente para o ficheiro de resultado, como se o tivesse escrito numa máquina de escrever, com todas as quebras de linha e espaços, sem que qualquer comando LaTeX seja executado.

Dentro de um parágrafo, um comportamento idêntico pode ser conseguido com

`\verb+`*texto*`+`

O sinal `+` é apenas um exemplo de um delimitador. Pode utilizar qualquer caracter excepto letras, `*`, ou um espaço. A maior parte dos exemplos LaTeX são escritos com este comando.

```
O comando \verb|\ldots| \ldots

\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

O comando `\ldots` . . .

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
A versão estrelada do
ambiente     verbatim
assinala os    espaços
que aparecem  no texto
\end{verbatim*}
```

```
A␣versão␣estrelada␣do
ambiente␣␣␣␣␣verbatim
assinala␣os␣␣␣␣espaços
que␣aparecem␣␣no␣texto
```

O comando `\verb` também pode ser estrelado:

```
\verb*|desta   forma :-) |
```

```
desta␣␣␣forma␣:-)␣
```

O ambiente `verbatim` e o comando `\verb` não devem ser utilizados dentro de parâmetros de outros comandos.

2.11.5 Tabelas

O ambiente `tabular` pode ser utilizado para imprimir lindas tabelas com linhas verticais e horizontais opcionais. O LaTeX determina a largura das colunas automaticamente.

O argumento *espec tabela* do comando

`\begin{tabular}[`*pos*`]{`*espec tabela*`}`

define o formato da tabela. Use um `l` para uma coluna de texto alinhado

à esquerda, `r` para texto alinhado à direita, e `c` para texto centrado; `p{`*`largura`*`}` para uma columa a conter texto com quebras de linha, e `|` para uma linha vertical.

O argumento *pos* especifica a posição vertical da tabela relativamente à linha base do texto envolvente. Use as letras `t`, `b` e `c` para especificar o alinhamento da tabela no topo, fundo ou ao centro.

Dentro de um ambiente `tabular`, o `&` salta para a próxima coluna, `\\` inicia uma nova linha e `\hline` insere uma linha horizontal. Pode adicionar linhas parciais usando `\cline{`*j*`-`*i*`}`, onde j e i são os números das colunas de onde e para onde a linha se deve estender.

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binário \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	hexadecimal
3700	octal
11111000000	binário
1984	decimal

```
\begin{tabular}{|p{4.7cm}|}
\hline
Bem-vindo ao parágrafo do Boxy.
Esperamos sinceramente que
se divirta com o espectáculo.\\
\hline
\end{tabular}
```

Bem-vindo ao parágrafo do Boxy. Esperamos sinceramente que se divirta com o espectáculo.

O separador de coluna pode ser especificado com o construtor `@{...}`. Este comando mata o espaço entre colunas e substitui-o com o que quer que esteja entre as chavetas. Um uso comum para este comando é explicado de seguida no problema de alinhamento de casas decimais. Outra aplicação possível é a supressão de espaços numa tabela com `@{}`.

```
\begin{tabular}{@{} l @{}}
\hline
sem espaço inicial\\
\hline
\end{tabular}
```

sem espaço inicial

```
\begin{tabular}{l}
\hline
espaço à esquerda e à direita\\
\hline
\end{tabular}
```

espaço à esquerda e à direita

Uma vez que não existe uma forma definida pelo LaTeX para alinhar colunas numericamente pelo ponto decimal,[15] pode fazer "batota" utilizando duas colunas: uma alinhada à direita com a parte inteira e uma alinhada à esquerda com a parte decimal. O comando `@{.}` substituí o espaço normal entre colunas com um ".", apenas, dando a aparência visual de uma única tabela alinhada pelo ponto decimal. Não se esqueça de substituir o ponto decimal nos números com o separador de coluna (`&`)! Um título para o par de colunas pode ser colocado acima da nossa "coluna" numérica usando o comando `\multicolumn`.

```
\begin{tabular}{c r @{.} l}
Expressão Pi       &
\multicolumn{2}{c}{Valor} \\
\hline
$\pi$               & 3&1416  \\
$\pi^{\pi}$         & 36&46   \\
$(\pi^{\pi})^{\pi}$ & 80662&7 \\
\end{tabular}
```

Expressão Pi	Valor
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{\textbf{Ene}} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Ene	
Mene	Muh!

Texto escrito no ambiente tabular fica sempre junto numa única página. Se quer escrever tabelas longas, possivelmente deverá dar uma olhadela ao pacote supertabular e ao longtabular.

2.12 Corpos Flutuantes

Hoje, a maior parte das publicações contém um grande número de figuras e tabelas. Estes elementos precisam de um tratamento especial, porque não podem ser partidos entre páginas. Uma maneira seria começar uma nova página sempre que uma figura ou uma tabela é demasiado grande para caber na página actual. Esta abordagem pode deixar páginas parcialmente vazias, o que fica muito mal.

A solução para este problema é fazer as figuras ou tabelas que não cabem na página actual"flutuar" para uma página mais tarde, enquanto se preenche a página actual com o texto seguinte. O LaTeX oferece dois ambientes para

[15] Se tiver o pacote de pacotes 'tools' instalado no sistema, dê uma olhadela ao pacote dcolumn.

corpos flutuantes; um para tabelas e um para figuras. Para ter todas as vantagens destes dois ambientes, é importante perceber de uma forma aproximada como o LaTeX maneja estas flutuações internamente. Doutro modo, estes elementos podem vir a ser uma grande fonte de frustração porque o LaTeX nunca os colocará onde os deseja.

Primeiro, vamos dar uma vista de olhos aos comandos que o LaTeX apresenta para corpos flutuantes:

Qualquer material incluso num ambiente `figure` ou `table` será tratado como uma matéria flutuante. Ambos os ambientes suportam parâmetros adicionais

`\begin{figure}[`*especificaçãode colocação*`]`

`\begin{table}[`*especificação de colocação*`]`

chamados de *especificação de colocação*. Estes parâmetros são usados para dizer ao LaTeX a localização para a qual o corpo flutuante se pode mover. A *especificação de colocação* é construída por um conjunto de caracteres de *permissões de colocação de corpos flutuantes*. Veja a tabela 2.4.

Note que `pt` e `em` são unidades TeX. Pode ler mais sobre unidades na tabela 5.5 na página 98.

Uma tabela pode ser iniciada com a seguinte linha

```
\begin{table}[!hbp]
```

A especificação de colocação `[!hbp]` indica ao LaTeX para colocar a tabela exactamente aqui (`h`) ou no fundo (`b`) de alguma página ou em alguma página especial para corpos flutuantes (`p`), e tudo isto mesmo que não fique muito bonito (`!`). Se nenhuma especificação for dada, é assumida a `[tbp]`.

Tabela 2.4: Permissões de Colocação de Corpos Flutuantes.

Espec.	Permissão para colocar em...
`h`	*here* — *aqui* neste exacto local onde ocorreu no meio do texto. É útil para pequenos objectos.
`t`	no *topo* da página
`b`	no *fundo* (*bottom*) da página
`p`	numa *página* especial apenas com corpos flutuantes..
`!`	sem considerar a maior parte do parâmetros internos[a] que podem fazer com que o corpo flutuante não seja colocado.

[a]Como o número máximo destes objectos permitidos por página.

O LaTeX irá colocar cada corpo flutuante de acordo com a especificação de colocação dada pelo autor. Se um destes não pode ser colocado na página actual, será enviado para a fila de espera correspondente[16] (para figuras ou tabelas). Quando uma nova página é iniciada, o LaTeX verifica primeiro se é possível preencher uma página especial de 'flutuações' com os objectos da lista de espera. Se isto não é possível, o primeiro objecto de cada uma das filas de espera são tratados como se tivessem ocorrido naquele momento no texto: o LaTeX tenta de novo colocar de acordo com a respectiva especificação de colocação (excepto o 'h,' que não volta a ser válido). Qualquer objecto que ocorra no texto será enviado para a fila de espera. O LaTeX mantêm estritamente a ordem original em que aparecem cada um dos tipos de objectos flutuantes. Esta é a razão pela qual uma figura que não pode ser colocada empurra todas as outras figuras para o fim do documento. Portanto:

> Se o LaTeX não está a colocar os objectos flutuantes como esperava é normal que seja apenas um deles numa das filas de espera que está a empatar serviço!

Enquanto que é possível dar ao LaTeX especificadores de colocação únicos, isto causa problemas. Se o objecto não cabe na localização pretendida, fica preso, a bloquear os objectos seguintes. Em particular, nunca deve utilizar apenas a opção `[h]`—esta é tão má que em versões recentes do LaTeX é automaticamente substituída por `[ht]`.

Depois de ter explicado a parte difícil, aqui estão mais algumas coisas a mencionar sobre os ambientes `table` e `figure`. Com o comando

`\caption{`*legenda*`}`

pode definir uma legenda para o objecto. Um número será automaticamente criado juntamente com o texto "Figure" ou "Table" e adicionado no início da legenda.

Os dois comandos

`\listoffigures` e `\listoftables`

funcionam de forma análoga ao comando `\tableofcontents`, imprimindo uma lista de figuras ou tabelas, respectivamente. Nestas listas, a legenda completa será repetida. Se tem tendência a usar grandes legendas, deve definir uma versão mais curta para as listas. Isto pode ser feito introduzindo a versão mais pequena entre parêntesis rectos depois do comando `\caption`.

```
\caption[Pequeno]{Loooooooooooooooooongoooooooooooo}
```

[16]Estas filas são FIFO—'first in first out', o primeiro a entrar é o primeiro a sair

Com `\label` e `\ref`, pode criar uma referencia para o corpo flutuante no meio do texto.

O seguinte exemplo desenha um quadrado e insere-o no documento. Pode usar isto se desejar reservar espaço para imagens que vai colar no documento pronto.

```
Figura~\ref{branco} é um exemplo de Arte-Pop.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Cinco por Cinco em Centimetros.} \label{branco}
\end{figure}
```

No exemplo anterior, o LaTeX irá tentar *mesmo* (`!`) colocar a figura justamente *aqui* (`h`).[17] Se isto não é possível, então tenta colocar a figura no *fundo* (`b`) da página. Se falhar a colocação da figura na página actual, determina se é possível criar uma página de corpos flutuantes que contenha esta figura e possivelmente algumas tabelas da fila de espera respectiva. Se não existir material suficiente para esta página especial, o LaTeX inicia uma nova página, e uma vez mais trata-a como se tivesse ocorrido nesse momento no texto.

Em algumas circunstâncias, pode ser necessário usar o comando

`\clearpage` ou mesmo o `\cleardoublepage`

que ordenam o LaTeX a colocar imediatamente no documento todos os objectos restantes das filas de espera e depois iniciar uma nova página. O `\cleardoublepage` obriga a começar a página do lado direito.

Irá aprender a incluir desenhos PostScript no seu documento LaTeX 2_ε mais tarde nesta introdução.

[17] assumindo que a fila de espera de figuras está vazia.

2.13 Protegendo Comandos Frágeis

O texto dado como argumento de comandos como `\caption` ou `\section` podem aparecer mais do que uma vez no documento (exemplo, na tabela de conteúdos assim como no corpo do documento). Alguns comandos falham quando usados no argumento de outros comandos do tipo `\section`. Estes são chamados comandos frágeis. Por exemplo, os comandos `\footnote` ou `\phantom` são frágeis. O que estes comandos precisam para funcionar é de protecção (não precisamos todos?). Pode protege-los usando o comando `\protect` antes deles.

O `\protect` apenas se refere ao comando que se lhe segue, nem sequer ao seu argumento. Em muitos casos, um `\protect` supérfluo não magoará ninguém.

```
\section{Estou a considerar
         \protect\footnote{proteger a minha nota}}
```

Capítulo 3

Fórmulas Matemáticas

Agora está pronto! Neste capítulo, vamos atacar a força principal do TEX: escrita de matemática. Mas, fique avisado: este capítulo apenas arranha este tópico superficialmente. Enquanto que as coisas explicadas aqui são suficientes para muitas pessoas, não desespere se não consegue encontrar uma solução para a escrita da fórmula que precisa. Mas, quase de certeza, que o seu problema está resolvido no AMS-LATEX[1]

3.1 Generalidades

O LATEX tem um modo especial para escrever matemática. Esta pode ser escrita directamente num parágrafo, ou o parágrafo pode ser quebrado para a desenhar à parte. O texto matemático *dentro* de um parágrafo é introduzido entre `\(` e `\)`, entre `$` e `$`, ou entre `\begin{math}` e `\end{math}`.

```
Adicione $a$ ao quadrado e $b$
ao quadrado para obter $c$ ao
quadrado. Ou, usando uma
fórmula matemática:
$c^{2}=a^{2}+b^{2}$
```

Adicione a ao quadrado e b ao quadrado para obter c ao quadrado. Ou, usando uma fórmula matemática: $c^2 = a^2 + b^2$

```
\TeX{} é pronunciado como
\(\tau\epsilon\chi\).\\[6pt]
100~m$^{3}$ de água\\[6pt]
Isto vem do meu
\begin{math}\heartsuit\end{math}
```

TEX é pronunciado como $\tau\epsilon\chi$.

100 m^3 de água

Isto vem do meu $\heartsuit$

[1] A *Sociedade Americana de Matemáticos - AMS* produziu algumas extensões poderosas do LATEX. Muitos dos exemplos deste capítulo fazem uso dessas extensões. Este pacote está disponível em todas as distribuições recentes do TEX. Se na sua está a faltar, vá a `CTAN:/tex-archive/macros/latex/required/amslatex`.

Quando quer as suas equações matemáticas maiores ou que sejam mostradas à parte do resto do parágrafo, é preferível *mostrá-las*, em vez de quebrar manualmente o parágrafo. Para fazer isto, pode delimitar a fórmula com `\[` e `\]`, ou entre `\begin{displaymath}` e `\end{displaymath}`.

```
Adicione $a$ ao quadrado e $b$
ao quadrado para obter $c$ ao
quadrado. Ou, usando uma
fórmula matemática:
\begin{displaymath}
c^{2}=a^{2}+b^{2}
\end{displaymath}
ou pode escrever menos com:
\[a+b=c\]
```

Adicione a ao quadrado e b ao quadrado para obter c ao quadrado. Ou, usando uma fórmula matemática:

$$c^{2}=a^{2}+b^{2}$$

ou pode escrever menos com:

$$a+b=c$$

Se desejar que o LaTeX numere as suas equações, pode usar o ambiente `equation`. Com este ambiente pode dar um nome (`\label`) ao número de uma equação para mais tarde referir-se a ela, noutro sítio qualquer do texto, usando `\ref` ou `\eqref`:

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
De (\ref{eq:eps}), descobrimos que
\ldots{}De \eqref{eq:eps} podemos
fazer o mesmo.
```

$$\epsilon > 0 \tag{3.1}$$

De (3.1), descobrimos que . . . De (3.1) podemos fazer o mesmo.

Note a diferença de estilo tipográfico entre equações que são inseridas no texto e as que são "mostradas":

```
$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$
```

$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \to \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

Existem diferenças entre o *modo matemático* e o *modo de texto*. Por exemplo, no *modo matemático*:

1. A maior parte dos espaços e mudanças de linha não têm qualquer significado, visto que todos os espaços ou são determinados logicamente da expressão matemática ou têm de ser especificados utilizando comandos tais como `\,`, `\quad` ou `\qquad`.

2. Linhas em branco não são permitidas. Apenas um parágrafo por fórmula.

3. Cada letra é considerada como sendo o nome de uma variável e será escrita como uma. Se quer escrever um texto normal dentro de uma fórmula (texto verticalmente direito com espaçamento normal) então tem de introduzir o texto usando o comando `\textrm{...}` (veja também a secção 3.7 na página 57).

```
\begin{equation}
\forall x \in \mathbf{R}:
\qquad x^{2} \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \qquad x^{2} \geq 0 \tag{3.2}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{para todo }x\in\mathbf{R}
\end{equation}
```

$$x^{2} \geq 0 \qquad \textrm{para todo } x \in \mathbf{R} \tag{3.3}$$

Os matemáticos podem ser muito esquisitos com os símbolos que usam: convencionalmente usa-se 'blackboard bold', que são obtidos usando `\mathbb` do pacote amsfonts ou amssymb. O último exemplo pode ser traduzido em

```
\begin{displaymath}
x^{2} \geq 0\qquad
\textrm{para todo }x\in\mathbb{R}
\end{displaymath}
```

$$x^{2} \geq 0 \qquad \textrm{para todo } x \in \mathbb{R}$$

3.2 Agrupar em Modo Matemático

A maior parte dos comandos de modo matemático funcionam apenas em relação ao caracter seguinte. Por isso, se quer que um comando afecte vários caracteres, tem de os agrupar usando chavetas: `{...}`.

```
\begin{equation}
a^x+y \neq a^{x+y}
\end{equation}
```

$$a^x+y \neq a^{x+y} \tag{3.4}$$

3.3 Construindo Blocos de Fórmulas Matemáticas

Nesta secção serão descritos os comandos mais importantes usados em impressões matemáticas. Dê uma olhadela à secção 3.10 na página 60 para uma lista detalhada de comandos para escrever símbolos matemáticos.

As **letras gregas minúsculas** são introduzidas como `\alpha`, `\beta`, `\gamma`, …, as maiúsculas, são introduzidas como `\Gamma`, `\Delta`, …[2]

```
$\lambda,\xi,\pi,\mu,\Phi,\Omega$
```

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

Expoentes e Índices podem ser especificados usando o caracter `^` e `_`.

```
$a_{1}$ \qquad $x^{2}$ \qquad
$e^{-\alpha t}$ \qquad
$a^{3}_{ij}$\\
$e^{x^2} \neq {e^x}^2$
```

$a_{1} \qquad x^{2} \qquad e^{-\alpha t} \qquad a^{3}_{ij}$
$e^{x^2} \neq {e^x}^2$

A **raíz quadrada** é introduzida como `\sqrt`; a raíz n^{th} é gerada com `\sqrt[`n`]`. O tamanho do sinal de raíz é determinado automaticamente pelo LaTeX. se apenas precisa do símbolo, deve utilizar `\surd`.

```
$\sqrt{x}$ \qquad
$\sqrt{ x^{2}+\sqrt{y} }$
\qquad $\sqrt[3]{2}$\\[3pt]
$\surd[x^2 + y^2]$
```

$\sqrt{x} \qquad \sqrt{ x^{2}+\sqrt{y} } \qquad \sqrt[3]{2}$
$\surd[x^2 + y^2]$

Os comandos `\overline` e `\underline` criam **linhas horizontais** directamente por cima ou por baixo de uma expressão.

```
$\overline{m+n}$
```

$\overline{m+n}$

Os comandos `\overbrace` e `\underbrace` criam **chavetas horizontais** longas por cima ou por baixo de uma expressão.

```
$\underbrace{ a+b+\cdots+z }_{26}$
```

$\underbrace{ a+b+\cdots+z }_{26}$

Para adicionar acentos matemáticos, tais como pequenas setas ou sinais til a variáveis, pode usar os comandos dados na tabela 3.1 na página 60. Chapéus largos e tils a cobrir vários caracteres são gerados com `\widetilde` e `\widehat`. O símbolo `'` dá um caracter de derivada.

```
\begin{displaymath}
y=x^{2}\qquad y'=2x\qquad y''=2
\end{displaymath}
```

$$y=x^{2}\qquad y'=2x\qquad y''=2$$

[2] Não existe um Alpha definido em LaTeX 2ε porque é idêntico a um A normal. Logo que a nova codificação para a matemática esteja pronta, as coisas irão mudar.

Vectores são especificados frequentemente adicionando um símbolo seta no topo das variáveis. Isto é feito com o comando `\vec`. Os dois comandos `\overrightarrow` e `\overleftarrow` são úteis para denotar os vectores de A para B.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec a\quad\overrightarrow{AB}$$

Usualmente não se escreve explicitamente o sinal ponto para indicar a operação de multiplicação. No entanto, é escrito para ajudar os olhos do leitor a agrupar uma fórmula. Deve usar `\cdot` nestes casos:

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2
    {\tau}_1 \cdot {\tau}_2
\end{displaymath}
```

$$v = {\sigma}_1 \cdot {\sigma}_2 {\tau}_1 \cdot {\tau}_2$$

Nomes de funções do estilo *logaritmo* são escritos frequentemente num tipo de letra vertical e não em itálico como as variáveis. Desta forma o LaTeX tem os seguintes comandos para escrever os nomes das funções mais importantes:

`\arccos`	`\cos`	`\csc`	`\exp`	`\ker`	`\limsup`	`\min`	`\sinh`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`	`\Pr`	`\sup`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`	`\sec`	`\tan`
`\arg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`	`\sin`	`\tanh`

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}=1$$

Para a função módulo, existem dois comandos: `\bmod` para o operador binário "$a \bmod b$" e `\pmod` para expressões como "$x \equiv a \pmod{b}$."

```
$a\bmod b$\\
$x\equiv a \pmod{b}$
```

$a\bmod b$
$x\equiv a \pmod{b}$

Uma **fracção** é escrita com o comando `\frac{...}{...}`. Frequentemente a forma 1/2 é preferível, porque fica melhor para pequenas porções de "material fraccionado".

```
$1\frac{1}{2}$~horas
\begin{displaymath}
\frac{ x^{2} }{ k+1 }\qquad
x^{ \frac{2}{k+1} }\qquad
x^{ 1/2 }
\end{displaymath}
```

$1\frac{1}{2}$ horas

$$\frac{ x^{2} }{ k+1 }\qquad x^{ \frac{2}{k+1} }\qquad x^{ 1/2 }$$

Para escrever coeficientes binomiais ou estruturas similares, pode usar o comando `\binom` do pacote `amsmath`.
`\newcommand{\newatop}[2]{\genfrac{}{}{0pt}{1}{#1}{#2}}`.)

```
\begin{displaymath}
\binom{n}{k}\qquad\mathrm{C}_n^k
\end{displaymath}
```

$$\binom{n}{k}\qquad\mathrm{C}_n^k$$

Para relações binárias pode ser útil colocar símbolos uns por cima dos outros. O `\stackrel` colocar o símbolo dado no primeiro argumento no tamanho de expoente sobre o segundo, que é escrito na sua posição usual.

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

O **integral** é gerado com `\int`, o **somatório** com `\sum` e o **produtório** com `\prod`. Os limites superiores e inferiores são especificados com `^` e `_`, tal como os sub-escritos e os super-escritos. [3]

```
\begin{displaymath}
\sum_{i=1}^{n} \qquad
\int_{0}^{\frac{\pi}{2}} \qquad
\prod_\epsilon
\end{displaymath}
```

$$\sum_{i=1}^{n} \qquad \int_{0}^{\frac{\pi}{2}} \qquad \prod_\epsilon$$

Para ter maior controlo sobre a colocação de índices em expressões complexas, o pacote `amsmath` providencia duas ferramentas adicionais: o comando `\substack` e o ambiente `subarray`:

```
\begin{displaymath}
\sum_{\substack{0<i<n \\ 1<j<m}}
   P(i,j) =
\sum_{\begin{subarray}{l} i\in I\\
         1<j<m
      \end{subarray}}     Q(i,j)
\end{displaymath}
```

$$\sum_{\substack{0<i<n \\ 1<j<m}} P(i,j) = \sum_{\begin{subarray}{l} i\in I\\ 1<j<m \end{subarray}} Q(i,j)$$

O TeX coloca à disposição todo o tipo de símbolos para **parêntesis** e outros delimitadores (por exemplo $[\ \langle\ \|\ \updownarrow$). Parêntesis curvos e rectangulares podem ser introduzidos nas teclas correspondentes e as chavetas com `\{`, mas todos os outros delimitadores são gerados com comandos especiais (por exemplo `\updownarrow`). Para uma lista de todos os delimitadores disponíveis, verifique a tabela 3.8 na página 62.

[3] AMS-LaTeX tem super-/sub-escritos multi-linha

```
\begin{displaymath}
{a,b,c}\neq\{a,b,c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

Se colocar o comando `\left` antes do delimitador que abre ou um `\right` antes do delimitador que fecha, o TEX irá determinar automaticamente o tamanho correcto do delimitador. Note que deve fechar cada um `\left` com um `\right` correspondente, e que o tamanho é determinado correctamente se ambos forem escritos na mesma linha. Se não quer um símbolo à direita, use o delimitador invisível '`\right.`'!

```
\begin{displaymath}
1 + \left( \frac{1}{ 1-x^{2} }
    \right) ^3
\end{displaymath}
```

$$1 + \left(\frac{1}{1-x^{2}} \right)^3$$

Em alguns casos é necessário especificar o tamanho correcto de um delimitador matemático à mão, o que pode ser feito usando os comandos `\big`, `\Big`, `\bigg` e `\Bigg` como prefixos para a maior partes dos delimitadores.[4]

```
$\Big( (x+1) (x-1) \Big) ^{2}$\\
$\big(\Big(\bigg(\Bigg($\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$\Big((x+1) (x-1) \Big) ^{2}$
$\big(\Big(\bigg(\Bigg($ $\quad$ $\big\}\Big\}\bigg\}\Bigg\}$ $\quad$ $\big\|\Big\|\bigg\|\Bigg\|$

Existem vários comandos para introduzir **três pontos** numa fórmula. `\ldots` escreve os pontos na linha base, `\cdots` põe-nos centrados. Além destes, existem os comandos `\vdots` para pontos verticais e `\ddots` para pontos diagonais. Pode encontrar outro exemplo na secção 3.5.

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

3.4 Espaçamento Matemático

Se os espaços do meio de fórmulas, escolhidos pelo TEX não são satisfatórios, podem ser ajustados inserindo comandos especiais. Existem alguns comandos para pequenos espaços: `\,` para $\frac{3}{18}$quad (␣), `\:` para $\frac{4}{18}$quad (␣) e `\;` para $\frac{5}{18}$ quad (␣). Os caracteres `\␣` geram um espaço de tamanho medio e `\quad`

[4] Estes comandos não funcionam como esperado se um comando de mudança de tamanho foi usado ou a opção `11pt` ou `12pt` foi especificada. Utilize o pacote exscale ou amsmath para corrigir este comportamento.

(␣) e `\qquad` (␣␣) produzem espaços largos. O tamanho de um `\quad` corresponde à largura do caracter 'M' no tipo de letra actual. O comando `\!` produz um espaço negativo de $-\frac{3}{18}$ quad (␣).

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_{D} g(x,y)
  \, \ud x\, \ud y
\end{displaymath}
em vez de
\begin{displaymath}
\int\int_{D} g(x,y)\ud x \ud y
\end{displaymath}
```

$$\int\!\!\!\int_{D} g(x,y)\, \mathrm{d} x\, \mathrm{d} y$$

em vez de

$$\int\int_{D} g(x,y)\mathrm{d} x \mathrm{d} y$$

Note que 'd' no diferencial é convencionalmente escrito em tipo de letra romana.

O AMS-LaTeX tem outras formas de controlar precisamente o espaço entre vários sinais de integral, nomeadamente os comando `\iint`, `\iiint`, `\iiiint`, e `\idotsint`. Com o pacote amsmath carregado, o exemplo acima seria escrito desta forma:

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_{D} \, \ud x \, \ud y
\end{displaymath}
```

$$\iint_{D} \, \mathrm{d} x \, \mathrm{d} y$$

Veja o documento electrónico `testmath.tex` (distribuído com o AMS-LaTeX) ou o Capítulo 8 do *The LaTeX Companion* [3] para mais detalhes.

3.5 Material Alinhado Verticalmente

Para escrever **matrizes**, use o ambiente `array`. Funciona de uma forma similar ao ambiente `tabular`. O comando `\\` é usado para quebrar as linhas.

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \left(\begin{array}{ccc} x_{11} & x_{12} & \ldots \\ x_{21} & x_{22} & \ldots \\ \vdots & \vdots & \ddots \end{array} \right)$$

Este ambiente também pode ser usado para escrever expressões que têm um delimitador esquerdo grande, usando um "`.`" como delimitador invisível à direita (`\right`):

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
 a & \textrm{if $d>c$}\\
 b+x & \textrm{na manhâ}\\
 l & \textrm{no resto do dia}
  \end{array} \right.
\end{displaymath}
```

$$y = \left\{ \begin{array}{ll} a & \textrm{if } d>c \\ b+x & \textrm{na manhâ} \\ l & \textrm{no resto do dia} \end{array} \right.$$

Da mesma forma que dentro do ambiente `tabular` pode desenhar linhas, pode fazer o mesmo no ambiente `array`, por exemplo, separando as entradas de uma matriz:

```
\begin{displaymath}
\left(\begin{array}{c|c}
 1 & 2 \\
\hline
3 & 4
\end{array}\right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array}\right)$$

Para formulas que ocupam mais do que uma linha ou para sistemas de equações, pode usar o ambiente `eqnarray`, e `eqnarray*` em vez de `equation`. Com `eqnarray` cada linha fica com um número de equação. O `eqnarray*` não numera o que quer que seja.

Os ambientes `eqnarray` e `eqnarray*` funcionam como uma tabela de três colunas da forma `{rcl}`, onde a coluna do meio pode ser utilizada para o sinal de igual ou o de diferente, ou qualquer outro sinal que pareça adequado. O comando `\\` quebra as linhas.

```
\begin{eqnarray}
f(x) & = & \cos x     \\
f'(x) & = & -\sin x   \\
\int_{0}^{x} f(y)dy &
 = & \sin x
\end{eqnarray}
```

$$f(x) = \cos x \qquad (3.5)$$

$$f'(x) = -\sin x \qquad (3.6)$$

$$\int_{0}^{x} f(y)dy = \sin x \qquad (3.7)$$

Note que o espaço de cada lado do sinal de igual é bastante grande. Pode ser reduzido especificando `\setlength\arraycolsep{2pt}`, como no seguinte exemplo.

Equações longas não irão ser divididas automaticamente em pequenos pedaços. O autor tem de especificar onde as quebrar e o espaço que deve indentar. Os dois métodos seguintes são os mais comuns para obter este resultado.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x -\frac{x^{3}}{3!}
     +\frac{x^{5}}{5!}-{}
                    \nonumber\\
 & & {}-\frac{x^{7}}{7!}+{}\cdots
\end{eqnarray}}
```

$$\begin{aligned} \sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ &\quad - \frac{x^7}{7!} + \cdots \qquad (3.8) \end{aligned}$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1
     -\frac{x^{2}}{2!} +{} }
                    \nonumber\\
 & & {}+\frac{x^{4}}{4!}
     -\frac{x^{6}}{6!}+{}\cdots
\end{eqnarray}
```

$$\begin{aligned} &\cos x = 1 - \frac{x^2}{2!} + \\ &\qquad + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \qquad (3.9) \end{aligned}$$

O comando `\nonumber` faz com que o LaTeX não gere um número para a equação actual.

Pode tornar-se difícil obter equações verticalmente alinhadas a aparecer de uma forma correcta com estes métodos; o pacote amsmath tem um conjunto mais poderoso de alternativas. (veja os ambientes `align`, `flalign`, `gather`, `multiline` e `split`).

3.6 Fantasmas

Não conseguimos ver fantasmas, mas eles continuam a ocupar algum espaço nas mentes de um grande número de pessoas. LaTeX não é diferente. Podemos usar isto para alguns truques interessantes de espaçamento.

Quando se alinha texto verticalmente usando `^` e `_`, o LaTeX é, por vezes, demasiado prestável. Usando o comando `\phantom` pode reservar espaço para caracteres que não aparecem no resultado final. A forma mais fácil de compreender é reparar nos sequintes exemplos:

```
\begin{displaymath}
{}^{12}_{\phantom{1}6}\textrm{C}
\qquad \textrm{em oposição a} \qquad
{}^{12}_{6}\textrm{C}
\end{displaymath}
```

$${}^{12}_{6}\textrm{C} \qquad \textrm{em oposição a} \qquad {}^{12}_{6}\textrm{C}$$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k}
\qquad \textrm{em oposição a} \qquad
\Gamma_{ij}^{k}
\end{displaymath}
```

$$\Gamma_{ij}^{k} \qquad \textrm{em oposição a} \qquad \Gamma_{ij}^{k}$$

3.7 Tamanho da Matemática

No modo matemático, o TEX selecciona o tamanho de letra de acordo com o contexto. Os expoentes, por exemplo, ficam escritos num tipo de letra mais pequeno. Se deseja escrever parte de uma equação em letras tipo `roman`, não use o comando `\textrm` porque o mecanismo de mudança de tamanho não irá funcionar, visto que o `\textrm` muda temporariamente para o modo de texto. Em vez disto, use `\mathrm` para manter o mecanismo de mudança de tamanho activo. Mas preste atenção, `\mathrm` irá funcionar apenas em pequenos textos. Os espaços ainda não estão activos e os caracteres acentuados não funcionam.[5]

```
\begin{equation}
2^{\textrm{nd}} \quad
2^{\mathrm{nd}}
\end{equation}
```

$$2^{\textrm{nd}} \quad 2^{\mathrm{nd}} \tag{3.10}$$

Por vezes, precisa de dizer ao LATEX o tamanho correcto a utilizar. Em modo matemático, o tamanho é especificado com os quatro comandos:

`\displaystyle` (123), `\textstyle` (123), `\scriptstyle` ($\scriptstyle 123$) e `\scriptscriptstyle` ($\scriptscriptstyle 123$).

A mudança de estilos também afecta a forma como os limites são desenhados.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
 \frac{\displaystyle
   \sum_{i=1}^n(x_i-\overline x)
   (y_i-\overline y)}
  {\displaystyle\biggl[
 \sum_{i=1}^n(x_i-\overline x)^2
\sum_{i=1}^n(y_i-\overline y)^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathop{\mathrm{corr}}(X,Y)= \frac{\displaystyle \sum_{i=1}^n(x_i-\overline x) (y_i-\overline y)} {\displaystyle\biggl[\sum_{i=1}^n(x_i-\overline x)^2 \sum_{i=1}^n(y_i-\overline y)^2 \biggr]^{1/2}}$$

Este é um desses exemplos nos quais não precisamos de parêntesis maiores do que os normais `\left[ \right]`.

3.8 Teoremas, Leis, . . .

Ao escrever documentos matemáticos, provavelmente precisa de uma forma de escrever “Lemas”, “Definições”, “Axiomas” e estruturas similares. O LATEX

[5]O pacote AMS-LATEX (amsmath) faz com que o comando `\textrm` funciona com mudança de tamanho.

suporta isto com o comando

`\newtheorem{`*nome*`}[`*contador*`]{`*texto*`}[`*secção*`]`

O argumento *nome*, é uma pequena palavra chave usada para identificar o "teorema." Com o argumento *texto*, pode definir o nome actual do "teorema" que será impresso no documento final.

Os argumentos entre parêntesis rectos são opcionais. São ambos usados para especificar a numeração usada no "teorema". Com o argumento *contador* pode especificar o *nome* do ambiente "teorema". O novo "teorema" será então numerado com a sua própria sequência. O argumento *secção* permite especificar a unidade de secção com o qual queremos que o "teorema" seja numerado.

Depois de executar o comando `\newtheorem` no preâmbulo do documento, pode usar o seguinte comando durante o documento.

`\begin{`*nome*`}[`*texto*`]`
Este é o meu interessante teorema
`\end{`*nome*`}`

Isto deve ser teoria suficiente. Os exemplos seguintes deverão remover quaisquer dúvidas e mostrar que o ambiente `\newtheorem` é demasiado complexo para se compreender.

```
% definições para o preâmbulo
% do documento
\newtheorem{law}{Lei}
\newtheorem{jury}[law]{Júri}
%no documento
\begin{law} \label{law:box}
Não te escondas na caixa.
\end{law}
\begin{jury}[O décimo]
Podes ser tu! Portanto, cautela
e vê a lei~\ref{law:box}\end{jury}
\begin{law}Não, Não, Não\end{law}
```

Lei 1 *Não te escondas na caixa.*

Júri 2 (O décimo) *Podes ser tu! Portanto, cautela e vê a lei 1*

Lei 3 *Não, Não, Não*

O teorema "Júri" usa o mesmo contador que o teorema "Lei". Desta forma, obtém um número que está em sequência com a outra "Lei". O argumento em parêntesis rectos é usado para especificar um título ou algo parecido para o teorema.

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
Se existem dois ou mais
formas de fazer algo, e
uma delas pode resultar
numa catástrofe, então
alguém a irá fazer.\end{mur}
```

Murphy 3.8.1 *Se existem dois ou mais formas de fazer algo, e uma delas pode resultar numa catástrofe, então alguém a irá fazer.*

O teorema de "Murphy" fica com um número que é ligado ao número da secção actual. Também pode usar outra unidade, como por exemplo capítulo ou subsecção.

3.9 Símbolos Gordos

É ligeiramente difícil obter símbolos *bold* em LaTeX; isto é possivelmente intencional visto que tipógrafos amadores tendem a usá-los em demasia. O comando `\mathbf` muda as letras para *bold*, mas apenas as letras normais, enquanto que os símbolos matemáticos são normalmente itálicos. Existe o comando `\boldmath`, mas *só pode ser usado fora do modo matemático.* Também funciona para símbolos.

```
\begin{displaymath}
\mu, M \qquad \mathbf{M} \qquad
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

$$\mu, M \qquad \mathbf{M} \qquad \boldsymbol{\mu, M}$$

Note que a vírgula também está em *bold* o que pode não ser o desejado.

O pacote amsbsy (incluído pelo amsmath) faz isto muito mais facilmente porque incluí o comando `\boldsymbol`.

```
\begin{displaymath}
\mu, M \qquad
\boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \qquad \boldsymbol{\mu}, \boldsymbol{M}$$

3.10 Lista de Símbolos Matemáticos

As tabelas seguintes demonstram todos os símbolos normalmente acessíveis no *modo matemático*.

Para usar os símbolos listados nas tabelas 3.12–3.16, [6] o pacote amssymb deve ser carregado no preâmbulo do documento e os tipos de letra AMS devem estar instalados no sistema. Se o pacote AMS e tipos de letras não estão instaladas no seu sistema, visite `CTAN:/tex-archive/macros/latex/required/amslatex`. Pode encontrar ainda outra lista de símbolos mais detalhada em `CTAN:info/symbols/comprehensive`.

Tabela 3.1: Acentos Matemáticos.

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

Tabela 3.2: Letras Gregas Minúsculas.

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		

Tabela 3.3: Letras Gregas Maiúsculas.

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

[6] Estas tabelas são derivadas do `symbols.tex` por David Carlisle e mudados subsequentemente extensivamente como sugerido por Josef Tkadlex.

Tabela 3.4: Relações Binárias.

Pode negar cada um dos seguintes símbolos adicionando o comando `\not` como prefixo do símbolo em causa.

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` or `\le`	$\geq$	`\geq` or `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$	`\sqsubset` [a]	$\sqsupset$	`\sqsupset` [a]	$\Join$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a]Use o pacote latexsym para aceder a estes símbolos

Tabela 3.5: Operadores Binários.

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$\ast$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

Tabela 3.6: Operadores GRANDES.

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

Tabela 3.7: Setas.

$\leftarrow$	`\leftarrow` or `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` or `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$>$	`\Rightarrow`	$=>$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Leftarrow>$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\Leftarrow>$	`\iff` (bigger spaces)	$\leadsto$	`\leadsto` [a]

[a] Use o pacote latexsym para aceder a estes símbolos

Tabela 3.8: Delimitadores.

<table>
<tr><td>$($</td><td><code>(</code></td><td>$)$</td><td><code>)</code></td><td>$\uparrow$</td><td><code>\uparrow</code></td><td>$\Uparrow$</td><td><code>\Uparrow</code></td></tr>
<tr><td>$[$</td><td><code>[</code> or <code>\lbrack</code></td><td>$]$</td><td><code>]</code> or <code>\rbrack</code></td><td>$\downarrow$</td><td><code>\downarrow</code></td><td>$\Downarrow$</td><td><code>\Downarrow</code></td></tr>
<tr><td>$\{$</td><td><code>\{</code> or <code>\lbrace</code></td><td>$\}$</td><td><code>\}</code> or <code>\rbrace</code></td><td>$\updownarrow$</td><td><code>\updownarrow</code></td><td>$\Updownarrow$</td><td><code>\Updownarrow</code></td></tr>
<tr><td>$\langle$</td><td><code>\langle</code></td><td>$\rangle$</td><td><code>\rangle</code></td><td>$\vert$</td><td><code>|</code> or <code>\vert</code></td><td>$\Vert$</td><td><code>\|</code> or <code>\Vert</code></td></tr>
<tr><td>$\lfloor$</td><td><code>\lfloor</code></td><td>$\rfloor$</td><td><code>\rfloor</code></td><td>$\lceil$</td><td><code>\lceil</code></td><td>$\rceil$</td><td><code>\rceil</code></td></tr>
<tr><td>$/$</td><td><code>/</code></td><td>$\backslash$</td><td><code>\backslash</code></td><td></td><td><code>.</code> (dual. empty)</td><td></td><td></td></tr>
</table>

Tabela 3.9: Grandes Delimitadores.

$\lgroup$	`\lgroup`	$\rgroup$	`\rgroup`	$\lmoustache$	`\lmoustache`	$\rmoustache$	`\rmoustache`
$\arrowvert$	`\arrowvert`	$\Arrowvert$	`\Arrowvert`	$\bracevert$	`\bracevert`		

Tabela 3.10: Símbolos Sortidos.

$\dots$	\dots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$	\mho [a]	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$	\Box [a]	$\Diamond$	\Diamond [a]
$\bot$	\bot	$\top$	\top	$\angle$	\angle	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\clubsuit$	\clubsuit	$\spadesuit$	\spadesuit
$\neg$	\neg or \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

[a]Use o pacote latexsym para aceder a este símbolo

Tabela 3.11: Símbolos não Matemáticos.

Estes símbolos também podem ser usados em modo texto.

†	\dag	§	\S	©	\copyright	®	\textregistered
‡	\ddag	¶	\P	£	\pounds	%	\%

Tabela 3.12: Delimitadores AMS.

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\lrcorner
$\lvert$	\lvert	$\rvert$	\rvert	$\lVert$	\lVert	$\rVert$	\rVert

Tabela 3.13: Letras AMS gregas e hebraicas.

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\daleth$	\daleth	$\gimel$	\gimel

Tabela 3.14: Relações Binárias AMS.

$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\doteqdot$	\doteqdot or \Doteq
$\leqslant$	\leqslant	$\geqslant$	\geqslant	$\risingdotseq$	\risingdotseq
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\fallingdotseq$	\fallingdotseq
$\leqq$	\leqq	$\geqq$	\geqq	$\eqcirc$	\eqcirc
$\lll$	\lll or \llless	$\ggg$	\ggg or \gggtr	$\circeq$	\circeq
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim	$\triangleq$	\triangleq
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox	$\bumpeq$	\bumpeq
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless	$\Bumpeq$	\Bumpeq
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\thicksim$	\thicksim
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\thickapprox$	\thickapprox
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\approxeq$	\approxeq
$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc	$\backsim$	\backsim
$\precsim$	\precsim	$\succsim$	\succsim	$\backsimeq$	\backsimeq
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Vdash$	\Vdash
$\Subset$	\Subset	$\Supset$	\Supset	$\Vvdash$	\Vvdash
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\backepsilon$	\backepsilon
$\therefore$	\therefore	$\because$	\because	$\varpropto$	\varpropto
$\shortmid$	\shortmid	$\shortparallel$	\shortparallel	$\between$	\between
$\smallsmile$	\smallsmile	$\smallfrown$	\smallfrown	$\pitchfork$	\pitchfork
$\vartriangleleft$	\vartriangleleft	$\vartriangleright$	\vartriangleright	$\blacktriangleleft$	\blacktriangleleft
$\trianglelefteq$	\trianglelefteq	$\trianglerighteq$	\trianglerighteq	$\blacktriangleright$	\blacktriangleright

Tabela 3.15: Setas AMS.

$\dashleftarrow$	\dashleftarrow	$\dashrightarrow$	\dashrightarrow	$\multimap$	\multimap
$\leftleftarrows$	\leftleftarrows	$\rightrightarrows$	\rightrightarrows	$\upuparrows$	\upuparrows
$\leftrightarrows$	\leftrightarrows	$\rightleftarrows$	\rightleftarrows	$\downdownarrows$	\downdownarrows
$\Lleftarrow$	\Lleftarrow	$\Rrightarrow$	\Rrightarrow	$\upharpoonleft$	\upharpoonleft
$\twoheadleftarrow$	\twoheadleftarrow	$\twoheadrightarrow$	\twoheadrightarrow	$\upharpoonright$	\upharpoonright
$\leftarrowtail$	\leftarrowtail	$\rightarrowtail$	\rightarrowtail	$\downharpoonleft$	\downharpoonleft
$\leftrightharpoons$	\leftrightharpoons	$\rightleftharpoons$	\rightleftharpoons	$\downharpoonright$	\downharpoonright
$\Lsh$	\Lsh	$\Rsh$	\Rsh	$\rightsquigarrow$	\rightsquigarrow
$\looparrowleft$	\looparrowleft	$\looparrowright$	\looparrowright	$\leftrightsquigarrow$	\leftrightsquigarrow
$\curvearrowleft$	\curvearrowleft	$\curvearrowright$	\curvearrowright		
$\circlearrowleft$	\circlearrowleft	$\circlearrowright$	\circlearrowright		

Tabela 3.16: Relações Binárias Negadas e Setas AMS.

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

Tabela 3.17: Operadores Binários AMS.

$\dotplus$	\dotplus	$\centerdot$	\centerdot	$\intercal$	\intercal
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\Cup or \doublecup	$\Cap$	\Cap or \doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes	$\circledast$	\circledast
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge		

Tabela 3.18: AMS Sortidos.

$\hbar$	\hbar	$\hslash$	\hslash	$\Bbbk$	\Bbbk
$\square$	\square	$\blacksquare$	\blacksquare	$\circledS$	\circledS
$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle	$\complement$	\complement
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\Game$	\Game
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\angle$	\angle	$\measuredangle$	\measuredangle	$\sphericalangle$	\sphericalangle
$\diagup$	\diagup	$\diagdown$	\diagdown	$\backprime$	\backprime
$\nexists$	\nexists	$\Finv$	\Finv	$\varnothing$	\varnothing
$\eth$	\eth	$\mho$	\mho		

Tabela 3.19: Alfabeto Matemático.

Exemplo	Comando	Pacote Necessário
ABCdef	\mathrm{ABCdef}	
ABCdef	\mathit{ABCdef}	
$\mathnormal{ABCdef}$	\mathnormal{ABCdef}	
$\mathcal{ABC}$	\mathcal{ABC}	euscript with option mathcal
$\mathscr{ABC}$	\mathscr{ABC}	mathrsfs
$\mathfrak{ABCdef}$	\mathfrak{ABCdef}	eufrak
$\mathbb{ABC}$	\mathbb{ABC}	amsfonts or amssymb

Capítulo 4

Especialidades

Quando se prepara um grande documento, o LaTeX irá ajudá-lo com algumas potencialidades, tais como a geração de índices, manuseamento de bibliografia, e mais. Uma descrição muito mais completa das especialidades e melhoramentos possíveis com o LaTeX podem ser encontradas no *LaTeX Manual* [1] e no *The LaTeX Companion* [3].

4.1 Incluindo Gráficos EPS

O LaTeX providencia as facilidades básicas para trabalhar com corpos flutuantes como imagens ou gráficos, com os ambientes `figure` e `table`.

Existem várias formas de gerar gráficos com LaTeX básico ou com pacotes de extensão ao LaTeX. Infelizmente, muitos utilizadores acham difícil de compreender. Por isso, não será explicado neste manual. Para mais informações sobre este assunto consulte o *The LaTeX Companion* [3] e o *LaTeX Manual* [1].

Uma forma muito mais fácil de colocar gráficos num documento, é gerar a imagem com um programa especializado [1] e depois incluir os gráficos gerados no documento. Novamente, o LaTeX tem pacotes que oferecem muitas formas diferentes de fazer isto mas, nesta introdução, vamos apenas discutir sobre gráficos no formato Encapsulated PostScript (EPS), porque são bastante fáceis de usar. Para usar imagens no formato EPS, tem de ter uma impressora PostScrpt [2]

Um bom conjunto de comandos para inclusão de gráficos está disponível no pacote graphicx por D. P. Carlisle. Este pacote faz parte de uma família de pacotes chamada de “graphics”[3].

[1] Como o XFig, CorelDraw!, Freehand, Gnuplot, ...

[2] Outra possibilidade é utilizar o GhostScript para imprimir, que está disponível em `CTAN:/tex-archive/support/ghostscript`. Os utilizadores de Windows devem querer dar uma olhadela ao GSview.

[3] `CTAN:/tex-archive/macros/latex/required/graphics`

Assumindo que está a trabalhar com um sistema com uma impressora PostScript e com o pacote graphicx instalado, pode usar o seguinte guia passo por passo para incluir uma imagem no seu documento:

1. Exportar a imagem do seu programa de edição gráfica para o formato EPS. [4]

2. Carregue o pacote graphicx no preâmbulo do documento com

   `\usepackage[`*driver*`]{graphicx}`

   onde *driver* é o nome do seu programa de conversão de "dvi para postscript". O programa deste tipo mais usado é chamado de `dvips`. O nome deste programa é necessário, porque não existe nenhum padrão para incluir gráficos em TEX. Sabendo o nome deste *driver*, o pacote graphicx pode escolher o método correcto para inserir informação sobre o gráfico no ficheiro `.dvi`, de forma a que a impressora perceba e possa incluir correctamente o ficheiro `.eps`.

3. Use o comando

   `\includegraphics[`*chave*=*valor*, ...`]{`*ficheiro*`}`

   para incluir o *ficheiro* no seu documento. O parâmetro opcional aceita uma lista separada por vírgulas de *chaves* e *valores* associados. As *chaves* podem ser usadas para alterar a largura, altura e rotação do gráfico incluído. A tabela 4.1 mostra as chaves mais importantes.

Tabela 4.1: Nomes das Chaves para o Pacote graphicx.

`width`	aumenta/diminuí a imagem para a largura especificada
`height`	aumenta/diminuí a imagem para a altura especificada
`angle`	roda o gráfico no sentido contrário ao dos ponteiros dos relógios
`scale`	altera a escala da imagem

O seguinte exemplo clarificará as ideias:

[4] Se o seu programa não conseguir exportar no formato EPS, pode tentar instalar um *driver* de impressora (uma LaserWriter da Apple, por exemplo) e depois imprimir para um ficheiro utilizando este *driver*. Com alguma sorte este ficheiro irá estar em formato EPS. Note que um EPS não deve conter mais do que uma página. Alguns *drivers* podem estar configurados explicitamente para produzir o formato EPS.

```
\begin{figure}
\begin{center}
\includegraphics[angle=90, width=0.5\textwidth]{test}
\end{center}
\end{figure}
```

Este comando inclui o gráfico guardado no ficheiro `test.eps`. O gráfico é rodado *inicialmente* por um ângulo de 90 graus e *depois* alterado de forma a que a largura seja 0.5 vezes a largura de um parágrafo normal (`textwidth`). A altura é calculada de forma a manter a relação altura/largura. A largura e altura também podem ser especificadas em dimensões absolutas. Veja a tabela 5.5 na página 98 para mais informação. Se quer saber mais sobre este tópico, leia [8] e [11].

4.2 Bibliografia

Pode produzir uma bibliografia com o ambiente `thebibliography`. Cada entrada começa com

`\bibitem[`*etiqueta*`]{`*marca*`}`

A *marca* é para ser usada durante o documento para citar o livro ou artigo descrito na entrada da bibliografia.

`\cite{`*marca*`}`

Se não usar a opção *etiqueta*, as entradas serão numeradas automaticamente. O parâmetro colocado após o comando `\begin{thebibliography}` define quanto espaço deve ser reservado para o número ou etiqueta. No exemplo seguinte, `{99}` indica ao LaTeX para considerar que nenhum dos números dos ítems da bibliografia vão ser maiores do que 99.

```
Partl~\cite{pa} propôs
que se \ldots

\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Partl [1] propôs que se ...

Bibliografia

[1] H. Partl: *German TEX*, TUGboat Volume 9, Issue 1 (1988)

Para projectos maiores, quererá usar o BibTEX. O BibTEX é incluído na maior parte das distribuição TEX. Permite manter uma base de dados bibliográfica e depois extrair as referencias relevantes para algo citado no documento. A apresentação visual das bibliografias geradas pelo BibTEX é baseada num conceito de folhas de estilo que permite criar bibliografias utilizando um grande conjunto de formatos pré-definidos.

4.3 Indexar

Uma característica muito útil de muitos livros é o seu índice remissivo. Com o LATEX e o programa de suporte `makeindex`,[5] um índice pode ser gerado de uma forma bastante simples. Esta introdução, apenas explicará os comandos de geração básicos. Para uma explicação mais aprofundada, veja o *The LATEX Companion* [3].

Para activar a indexação do LATEX, deve incluir o pacote makeidx no preâmbulo do documento com

```
\usepackage{makeidx}
```

e os comandos especiais de indexação devem ser activados colocando o comando

```
\makeindex
```

no preâmbulo do documento.

[5] Em sistemas que não suportam necessariamente nomes de ficheiros com mais de oito caracteres, o programa pode ter o nome de `makeidx`.

Tabela 4.2: Exemplos da Sintaxe das Chaves de Indexação.

Exemplo	**Entrada no índice**	**Comentário**
`\index{olá}`	olá, 1	Entrada normal
`\index{olá!Pedro}`	Pedro, 3	Sub-entrada de 'olá'
`\index{Sam@\textsl{Sam}}`	*Sam*, 2	Entrada formatada
`\index{Lin@\textbf{Lin}}`	**Lin**, 7	O mesmo que a de cima
`\index{Jenny|textbf}`	Jenny, **3**	Número de página formatado
`\index{Joe|textit}`	Joe, *5*	O mesmo que a de cima
`\index{eolienne@\'eolienne}`	éolienne, 4	Uso de acentuação

O conteúdo do índice é especificado com comandos

`\index{`*chave*`}`

onde a *chave* é a entrada no índice. Insere os comandos de indexação nos pontos do texto para onde quer que o índice aponte. A tabela 4.2 explica a sintaxe do argumento *chave*, com vários exemplos.

Quando o ficheiro de origem é processado com o LaTeX, cada comando `\index` escreve a entrada apropriada com o número de página actual para um ficheiro especial. O ficheiro tem o mesmo nome do documento original, mas com uma extensão diferente (`.idx`). Este ficheiro `.idx` deve ser processado posteriormente com o comando `makeindex`.

`makeindex` *nomedoficheiro*

O programa `makeindex` gera um índice ordenado com o mesmo nome mas com a extensão `.ind`. Quando o documento for processado novamente com o LaTeX, este índice ordenado será incluído no documento no ponto onde o LaTeX encontrar

`\printindex`

O pacote showidx que vem com o LaTeX 2_ε imprime todas as entradas na margem esquerda do texto. Isto é muito útil para reler um documento e verificar o índice.

4.4 Cabeçalhos

O pacote fancyhdr,[6] escrito por Piet van Oostrum, fornece alguns comandos simples que permitem configurar o cabeçalho e o rodapé do seu documento. Se olhar para o topo desta página, irá ver uma aplicação possível deste pacote.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% com isto temos a certeza que os cabeçalhos do
% capítulo e secção são em minúsculas.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{}  % apagar as configurações actuais
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % fazer espaço para o risco
\fancypagestyle{plain}{%
   \fancyhead{} % Tirar cabeçalhos de página vazias
   \renewcommand{\headrulewidth}{0pt} % e o risco
}
```

Figura 4.1: Exemplo de configuração do fancyhdr.

O maior problema ao configurar cabeçalhos e rodapés é fazer coisas tais como colocar nomes de secções e capítulos aí. O LaTeX faz isto numa abordagem de dois níveis. Na definição da cabeçalho e do rodapé, usa os comandos `\rightmark` e `\leftmark` para representar a capítulo e a secção actual, respectivamente. Os valores destes dois comandos são reescritos sempre que um comando de capítulo ou secção é processado.

Para maior flexibilidade, o comando `\chapter` e os seus amigos não redefinem, eles mesmos, o `\rightmark` e `\leftmark`, mas chamam outros comandos (`\chaptermark`, `\sectionmark` ou `\subsectionmark`) que são responsáveis por redefinir `\rightmark` e `\leftmark`.

Assim, se quer mudar o aspecto do nome de capítulo na linha de cabeçalho, deve redefinir, simplesmente, o comando `\chaptermark`.

A figura 4.1 mostra uma configuração possível para o pacote fancyhdr que faz cabeçalhos idênticos aos deste livro. De qualquer maneira, a minha

[6]Disponível em `CTAN:/tex-archive/macros/latex/contrib/supported/fancyhdr`.

sugestão é que copie a documentação deste pacote a partir do endereço mencionado na nota de página.

4.5 O Pacote Verbatim

Anteriormente neste livro, foi explicado o ambiente `verbatim`. Nesta secção, vai aprender sobre o *pacote* verbatim. O pacote verbatim é basicamente uma re-implementação do ambiente `\verbatim` que contorna as limitações do ambiente original. Isto, por si, não é espectacular, mas com a implementação do pacote verbatim, existem novas funcionalidades e por essa razão menciono este pacote aqui. O pacote verbatim providencia o comando

`\verbatiminput{`*nomedoficheiro*`}`

que permite incluir texto ASCII puro no documento como se estivesse dentro do ambiente `verbatim`.

Como o pacote verbatim é parto do conjunto 'tools', deve o encontrar instalado em quase todos os sistemas. Se quer saber mais sobre este pacote, leia [9].

4.6 Instalando Pacotes LaTeX

A maior parte das instalações LaTeX vêm com um grande conjunto de pacotes já instalados, mas existem muitos mais disponíveis na Internet. O principal sítio para procurar por pacotes LaTeX é o CTAN (`http://www.ctan.org/`).

Pacotes, tais como o geometry ou o hyphenat, e muitos outros, são tipicamente constituídos de dois ficheiros: um com a extensão `.ins` e outro com a extensão `.dtx`. Muitas vezes irá existir um `readme.txt` com uma breve descrição do pacote. Deve, certamente, ler este ficheiro antes de qualquer outra coisa.

Em qualquer caso, uma vez copiados os ficheiros do pacote para a sua máquina, continua a precisar de os processar de forma a que (a) a sua distribuição TeX saiba alguma coisa sobre os novos pacotes e (b) obtenha a documentação. Aqui está como fazer a primeira parte:

1. Corra o LaTeX no ficheiro `.ins`. Isto vai extrair um ficheiro `.sty`.

2. Mova o ficheiro `.sty` para um sítio onde a sua distribuição o consiga encontrar. Normalmente, isto é na sua `.../`*`localtexmf`*`/tex/latex` subdirectoria (Utilizadores Windows devem sentir-se livres de mudar a direcção das barras).

3. Actualizar a base de dados da sua distribuição. O comando depende da distribuição de LaTeX que usa: teTeX, fpTeX – `texhash`; web2c –

`maktexlsr`; MikTeX – `initexmf -update-fndb` ou use um interface gráfico.

Agora pode extrair a documentação do ficheiro `.dtx`:

1. Corra o LaTeX no ficheiro `.dtx`. Isto irá gerar um ficheiro `.dvi`. Note que deve precisar de usar o LaTeX várias vezes antes de obter as referencias cruzadas correctamente;
2. Verifique se o LaTeX produziu um ficheiro `.idx` no meio dos outros vários ficheiros que tem. Se não encontra este ficheiro, então proceda para o passo 5.
3. Para gerar o índice, escreve o seguinte:
   `makeindex -s gind.ist` *`nome`*
   (onde *nome* é o nome do ficheiro principal sem a extensão).
4. Corra de novo o LaTeX no ficheiro `.dtx`.
5. Por fim, faça um ficheiro `.ps` ou `.pdf` para aumentar o seu prazer de leitura.

Um perigo final: muito raramente, pode encontrar um ficheiro `.glo` (glossário). Este é processado depois do passo 4 e antes do 5:

```
makeindex -s gglo.ist -o nome.gls nome.glo
```

Tenha a certeza de passar o LaTeX no ficheiro `.dtx` uma última vez antes de mover para o passo 5.

4.7 Usar o pdfLaTeX

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

O PDF é um formato de documentos de hipertexto. Como uma página da Internet, algumas palavras no documento são marcadas como hiper-ligações. Estas, ligam para outros sítios no documento ou mesmo para outros documentos. Se clicar numa destas ligações será transportado para o destino da mesma ligação. No contexto do LaTeX, isto significa que todas as ocorrências de `\ref` e `\pageref` passarão a hyper-ligações. Adicionalmente, a tabela de conteúdos, o índice remissivo e todas as estruturas deste género passarão a hiper-ligações.

A maior parte das páginas de Internet que se encontram nos dias que correm estão escritas em HTML *(HyperText Markup Language)*. Este formato tem duas desvantagens ao escrever documentos científicos:

1. Incluir fórmulas matemáticas em documentos HTML não é, geralmente, possível. Embora exista um standard para isso, a maior parte dos navegadores de hoje não a suportam, ou têm falta de tipos de letra.

2. Imprimir documentos HTML é possível mas o seu resultado varia de forma incontrolável entre plataformas e navegadores. Os resultados estão a milhas da qualidade que esperamos quando estamos habituados ao mundo LATEX.

Existem muitas tentativas de criar tradutores de LATEX para HTML. Alguns tiveram bastante sucesso no que respeita a serem capazes de produzir páginas de Internet legíveis a partir dum ficheiro LATEX comum. Mas todos eles cortam caminho para terem o trabalho feito. Assim que começar a usar características complexas do LATEX e pacotes externos, as coisas começarão a deixar de funcionar. Autores que querem preservar a qualidade tipográfica única dos seus documentos ao publicá-los na Internet, transformam-os em PDF *(Portable Document Format—Formato de documentos portável)*, que preserva o formato do documento e que permite navegação de hipertexto. Quase todos os navegadores modernos dispõem de plugins que permitem ver correctamente documentos PDF.

Em contraste com os formados DVI e PS, os documentos PDS podem ser vistos e impressos em todas as plataformas modernas (Unix, Mac, Windows), graças ao programa Adobe Acrobat Reader, que pode ser copiado livremente da página de Internet da Adobe. Em muitos computadores, até existe pré-instalado com a distribuição base do sistema operativo.

4.7.1 Documentos PDF para a Internet

A criação de ficheiros PDF a partir do LATEX é muito simples, graças ao programa pdfTEX desenvolvido por Hàn Thế Thành. O pdfTEX produz um ficheiro PDF onde o normal TEX produz DVI. Também existe um pdfLATEX, que produz PDF directamente do LATEX.

Quer o pdfTEX quer o pdfLATEX são instalados automáticamente pela maior parte das distribuições modernas do TEX, como a teTEX, fpTEX, MikTEX, TEXLive e CMacTEX.

Para produzir um PDF em vez do DVI, é suficiente substituir o comando `latex file.tex` por `pdflatex file.tex`. Em sistemas em que o LATEX não é chamado da linha de comando, deverá ser necessário encontrar um botão especial no TEXControlCenter.

In LATEX you can define the the paper size with an optional documentclass argument such as `a4paper` or `letterpaper`. This works in pdfLATEX too, but on top of this pdfTEX also needs to know the physical size of the paper and not just the area to use for the layout. If you use the hyperref

package (see page 78), the papersize will be adjusted automatically. Otherwise you have to do this manually by putting the following lines into the preamble of the document:

```
\pdfpagewidth=\paperwidth
\pdfpageheight=\paperheight
```

A secção seguinte irá aprofundar em maior detalhe as diferenças entre o LaTeX normal e o pdfLaTeX. As maiores diferenças dizem respeito a três áreas: os tipos de letra a usar, o formato das imagens a incluir, e a configuração manual das hiper-ligações.

4.7.2 Os tipos de letra

pdfLaTeX pode funcionar com todos os tipos de letras (PK bitmaps, TrueType, PostScript type 1. . .) mas o formato principal do LaTeX, os tipos de letra bitmap PK, produzem resultados muito feios quando o documento é mostrado com o Acrobat Reader. É preferível usar os tipos de letra PostScript Type 1 exclusivamente para produzir documentos que sejam bem visíveis.

Ainda não falamos sobre tipos de letra neste livro porque o LaTeX manuseia-as por si próprio sem problemas, uma vez que usa o seu próprio conjunto de tipos de letras adaptados aos requisitos de publicação científica. Na verdade, existem dois conjuntos de tipos de letra TeX: *Computer Modern* (CM), que consistem em 128 caracteres, que é o tipo de letra por omissão, e *Extended Cork* (EC), que consiste em 256 caracteres. A especialidade do conjunto EC é que contém caracteres especiais para todos os caracteres acentuados específicos de certas línguas Europeias, como ä ou è. Isto permite que a hifenização funcione correctamente com palavras que contêm estes caracteres. Pode activar os caracteres EC colocando o comando `\usepackage[T1]{fontenc}` no preâmbulo do documento (ver página 27). Infelizmente, não existe um conjunto de tipos de letra PostScript type 1 gratuito. Felizmente, existem duas formas de batota.

- Pode colocar a linha `\usepackage{aeguill}` no preâmbulo do documento, para usar os tipos de letra virtuais AE.
- Ou pode usar `\usepackage{mltex}`, mas isto só funciona quando o seu pdfTeX foi compilado com a opção mltex.

O conjunto de tipos de letra AE, tal como o sistema MlTeX, faz com que o TeX acredite que tem um conjunto completo de 256 caracteres à sua disposição criando as letras que faltam a partir de caracteres disponíveis no tipo de letra CM normal, que existe numa variante PostScript type 1. Isto tem a grande vantagem de que a hifenização funciona bem para as línguas europeias. A única desvantagem desta abordagem é que os caracteres artificiais

AE não funcionam na função de procura (`Find`) do Acrobat Reader, pelo que não pode procurar por palavras que contenham caracteres acentuados no seu ficheiro PDF final.

Outra solução é não usar o conjunto de tipos de letra CM, mas mudar para outro tipo PostScript type 1. Na verdade, algumas até estão incluídas em cada cópia do Acrobat Reader. Porque estes tipos têm caracteres de tamanhos diferentes, o formato do texto na página irá mudar. Geralmente irá usar mais espaço do que com os tipos CM, que são bastante eficientes no que respeita ao espaçamento. Também, a coerência visual global do documento irá sofrer porque a Times, Helvetica e Courier (as candidatas primárias para o serviço de substituição) não foram desenhadas para funcionar em harmonia num único documento como foi feito para os tipos de letra Computer Modern.

Dois conjuntos de tipos prontos a usar estão disponíveis para este fim: pxfonts, que é baseado nas *Palatino*, como tipo para o texto do corpo, e o pacote txfonts, que é baseado no *Times*. Para os usar é suficiente colocar as seguintes linhas no preâmbulo do seu documento:

```
\usepackage[T1]{fontenc}
\usepackage{pxfonts}
```

Nota: irá encontrar linhas como

```
Warning: pdftex (file eurmo10): Font eurmo10 at ... not found
```

no ficheiro `.log` depois de compilar o seu ficheiro. Significam que algum tipo de letra usado no documento não foi encontrado. Tem realmente de resolver estes problemas, ou o PDF resultante pode *não mostrar as páginas com os caracteres em falta*.

Todo este negócio de tipos de letra, especialmente a falta de um bom conjunto de tipos de letra EC de qualidade equivalente às CM no formato type 1, está a ocupar as mentes de muita gente, e portanto, novas soluções estão a aparecer a todo o momento.

4.7.3 Uso de Gráficos

Incluir gráficos num documento funciona melhor com o pacote graphicx (ver página 67). Ao usar a opção especial `pdftex` como *driver*, o pacote irá funcionar com o pdfLaTeX de forma simples:

```
\usepackage[pdftex]{color,graphicx}
```

No exemplo acima, também incluí a opção *color*, visto que o uso de cores em documentos mostrados na Internet é bastante natural.

Chega de boas notícias. As más é que o EPS (Encapsulated Postscript), o seu formato favorito para gráficos em LaTeX, *não* funciona para ficheiros

PDF. Se não definir uma extensão no nome do ficheiro do comando `\includegraphics`, o pacote graphicx irá procurar um ficheiro que lhe seja adequado, dependendo das opções do *driver*. Para `pdftex` irá procurar os formatos `.png`, `.pdf`, `.jpg`, `.mps` (MetaPost), e `.tif`—mas *não* `.eps`.

A forma simples de resolver este problema é simplesmente converter os seus ficheiros EPS para o formato PDF usando a utilidade `epstopdf` existente em tantos sistemas. Para gráficos vectoriais (desenhos) esta é uma grande solução. Para mapas de bits (fotografias) isto pode não ser ideal, porque o formato PDF suporta nativamente a inclusão de imagens PNG e JPEG. PNG é bom para imagens de aplicações e outras imagens com poucas cores. O JPEG é bom para fotografias, visto ser eficiente em relação ao espaço ocupado em disco.

Até pode ser desejável não desenhar algumas figuras geométricas mas descrevê-las com uma linguagem especializada, como o MetaPost, que pode ser encontrada em quase todas as distribuições do TeX, e vem com o seu próprio manual extensivo.

4.7.4 Ligações de Hipertexto

O pacote hyperref irá ter o cuidado de transformar todas as referências internas do seu documento em hiper-ligações. Para que isto funcione devidamente, alguma magia é necessário, e portanto, tem de colocar `\usepackage[pdftex]{hyperref}` como *último* comando do preâmbulo do seu documento.

Muitas opções estão disponíveis para configurar o comportamento do pacote hyperref:

- como uma lista separada por vírgulas depois da opção pdftex
  `\usepackage[pdftex]{hyperref}`
- ou em linhas individuais com o comando `\hypersetup{`*opções*`}`.

A única opção necessária é `pdftex`; as outras são opcionais e permitem alterar o comportamento por omissão do hyperref.[7] Na lista seguinte, os valores por omissão estão escritos num tipo de letra vertical (portanto, não itálico).

`bookmarks (=true,`***false***`)` mostra (true) ou esconde a barra de bookmarks ao visualizar o documento;

`pdftoolbar (=true,`***false***`)` mostra ou esconde a barra de ferramentas do Acrobat;

[7] É de notar que o pacote hyperref não está limitado ao uso com o pdfTeX. Também pode ser configurado para embeber informação específica do PDF no ficheiro DVI, resultado normal do LaTeX, que depois irá colocar no ficheiro PS com o `dvips` e que, finalmente, será usado pelo Adobe Distiller quando se tentar converter de PS para PDF.

`pdfmenubar` (=true,***false***) mostra ou esconde o menu do Acrobat;

`pdffitwindow` (=true,***false***) ajusta, ou não, automaticamente o tamanho inicial do texto quando visualizado;

`pdftitle` (={texto}) define o título que será mostrado na janela `Document Info` do Acrobat;

`pdfauthor` (={texto}) o nome do autor do PDF;

`pdfnewwindow` (=true,***false***) define se uma nova janela deve ser aberta quando uma ligação envia para fora do documento actual;

`colorlinks` (=true,***false***) mostra ou não as ligações a cor. As cores destas ligações podem ser configuradas com as seguintes opções:

- `linkcolor` (=color,***red***) cor de ligações internas (secções, páginas, etc),
- `citecolor` (=color,***green***) cor de citações (bibliografia),
- `filecolor` (=color,***magenta***) cor de ligações para ficheiros;
- `urlcolor` (=color,***cyan***) cor de ligações de URL (mail, web).

Se estiver contente com os valores por omissão, use

```
\usepackage[pdftex]{hyperref}
```

Para ter a lista de bookmarks aberta e ligações em cor: (os valores `=true` são opcionais):

```
\usepackage[pdftex,bookmarks,colorlinks]{hyperref}
```

Ao criar PDFs destinados a serem impressos, as ligações coloridas não são boa ideia uma vez que irão acabar cinzentas no resultado final, tornando-se difíceis de ler:

```
\usepackage{hyperref}
\hypersetup{colorlinks,%
            citecolor=black,%
            filecolor=black,%
            linkcolor=black,%
            urlcolor=black,%
            pdftex}
```

Quando quer apenas providenciar informação para a secção de informação sobre o documento PDF:

```
\usepackage[pdfauthor={Pierre Desproges}%
            pdftitle={Des femmes qui tombent},%
            pdftex]{hyperref}
```

Além das hiper-ligações automáticas para referências cruzadas, é possível embeber explicitamente ligações usando

`\href{`*url*`}{`*texto*`}`

O código

```
O endereço do \href{http://www.ctan.org}{CTAN}.
```

produz o resultado "CTAN"; um clique na palavra "CTAN" irá levá-lo à página de Internet do CTAN.

Se o destino da ligação não é um URL mas um ficheiro local, pode usar o comando `\href` da seguinte forma:

```
O documento completo está \href{manual.pdf}{aqui}
```

Que produz o texto "O documento completo está aqui". Ao clicar na palavra "aqui" irá abrir o ficheiro `manual.pdf`. (O nome do ficheiro é relativo à localização actual do documento actual).

O autor de um artigo pode desejar que os seus leitores enviem-lhe mensagens de correio electrónico usando o comando `\href` dentro do comando `\author` na página principal do documento:

```
\author{Mary Oetiker $<$\href{mailto:mary@oetiker.ch}%
       {mary@oetiker.ch}$>$
```

Note que coloquei a ligação de forma a que o meu endereço apareça não só na ligação mas também na página. Isso é importante porque a ligação
`\href{mailto:mary@oetiker.ch}{Mary Oetiker}`
irá funcionar bem dentro do Acrobat, mas assim que a página seja impressa o endereço nunca mais seria visível.

4.7.5 Problemas com Ligações

Mensagens como a seguinte:

```
! pdfTeX warning (ext4): destination with the same identifier
  (name{page.1}) has been already used, duplicate ignored
```

aparecem quando um contador é reinicializado, por exemplo, ao usar o comando `\mainmatter` providenciado pela classe de documento `book`. Este reinicializa o contador do número de página a 1 antes do primeiro capítulo do livro. Mas como o prefácio do livro também tem como número de página

o 1, todas as ligações à “página 1” deixarão de ser únicas, e daí que apareça o aviso de que o duplicado tenha sido ignorado.

A contra-medida consiste em colocar `plainpages=false` nas opções do hyperref. Isto, infelizmente, ajuda apenas com o contador de páginas. Uma solução ainda mais radical é o uso da opção `hypertexnames=false`, mas que irá causar as ligações de páginas deixar de funcionar.

4.7.6 Problemas com Bookmarks

O texto mostrado nas bookmarks não aparece sempre como esperava que aparecessem. Porque as bookmarks são “apenas texto,” muito menos caracteres estão disponíveis do que para texto normal LaTeX. Hyperref irá detectar esses problemas e avisar:

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

Depois, pode contornar o problema providenciando um texto para a bookmark, que irá substituir o texto em causa:

`\texorpdfstring{`*Texto TEX*`}{`*Texto da Bookmark*`}`

As expressões matemáticas são candidatas especiais para este tipo de problema:

```
\section{\texorpdfstring{$E=mc^2$}%
        {E\ =\ mc\texttwosuperior}}
```

o que torna `\section{$E=mc^2$}` em “E=mc2” para a área de bookmark.

Mudanças de cor também não viajam para as bookmarks:

```
\section{\textcolor{red}{Red !}}
```

produz “redRed!”. O comando `\textcolor` será ignorado mas o seu argumento (red) será impresso.

Se usar

```
\section{\texorpdfstring{\textcolor{red}{Red !}}{Red\ !}}
```

o resultado irá ser muito mais legível.

Compatibilidade entre LaTeX e pdfLaTeX

Idealmente o seu documento deveria compilar igualmente bem com o LaTeX quer com o pdfLaTeX. O principal problema a este respeito é a inclusão a gráficos. A solução simples é para *deixar cair sistematicamente* a extensão do comando `\includegraphics`. Um formato adequado será automaticamente procurado na directoria actual. Tudo o que tem de fazer é criar versões

apropriadas dos ficheiros de imagens. O LaTeX irá procurar por ficheiros `.eps`, e pdfLaTeX irá tentar incluir um ficheiro com extensão `.png`, `.pdf`, `.jpg`, `.mps` ou `.tif` (por esta ordem).

Para os casos em que deseja usar código diferente para a versão PDF do seu documento, pode adicionar:

```
\newif\ifPDF
\ifx\pdfoutput\undefined\PDFfalse
\else\ifnum\pdfoutput > 0\PDFtrue
     \else\PDFfalse
     \fi
\fi
```

nas primeiras linhas do seu documento, que define um comando especial que irá permitir escrever de forma simples código condicional:

```
\ifPDF
  \usepackage[T1]{fontenc}
  \usepackage{aeguill}
  \usepackage[pdftex]{graphicx,color}
  \usepackage[pdftex]{hyperref}
\else
  \usepackage[T1]{fontenc}
  \usepackage[dvips]{graphicx}
  \usepackage[dvips]{hyperref}
\fi
```

No exemplo, incluí o pacote hyperref mesmo na versão não PDF. Isto faz com que o comando `\href` funcione em todos os casos, o que me poupa de alterar cada uma das suas ocorrências numa instrução condicional.

Note que em distribuições recentes do TeX (TeXLive, por exemplo), a escolha entre `pdftex` e `dvips` ao chamar o pacote graphicx e color irá acontecer automaticamente de acordo com as preferências escolhidas nos ficheiros de configuração `graphics.cfg` e `color.cfg`.

4.8 Criar Apresentações com pdfscreen

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

Pode apresentar os resultados do seu trabalho científico num quadro preto, com transparências, ou directamente do seu portátil usando algum programa de apresentações.

O pdfLaTeX combinado com o pacote pdfscreen permite criar apresentações em PDF, igualmente coloridas e vivas como as do *PowerPoint*, mas muito mais portáveis porque o Acrobat Reader existe em muitos mais sistemas.

A classe pdfscreen usa graphicx, color e hyperref com opções adaptadas a apresentações em écran.

```
\documentclass[pdftex,12pt]{article}
%%% algumas extensões %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage[latin1]{inputenc}
\usepackage[english]{babel}
\usepackage[T1]{fontenc}
\usepackage{aeguill}
%%% pdfscreen %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\usepackage[screen,panelleft,chocolate]{pdfscreen}
% Formato do écran
\panelwidth=25mm
%%          altura largura
\screensize{150mm}{200mm}
%%     esquerda direita topo fundo
\marginsize{42mm}{8mm}{10mm}{10mm}
% Cor ou imagem para o fundo
\overlayempty
\definecolor{mybg}{rgb}{1,0.9,0.7}
\backgroundcolor{mybg}
% Logotipo
\emblema{MyLogo}
%%% Para o PPower4 (pós-processador) %%%%%%%%%%%%%%%%%%
\usepackage{pause}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\begin{document}
\begin{slide}
\begin{itemize}
\item Boas notícias\dots \pause
\item Más notícias
\end{itemize}
\end{slide}
\end{document}
```

Figura 4.2: Exemplo dum ficheiro para o pdfscreen

Para criar este tipo de documentos irá trabalhar normalmente na classe `article`. A figura 4.2 mostra um exemplo de um ficheiro para o pdfscreen. Primeiro há que carregar o pacote pdfscreen juntamente com as opções apropriadas:

`screen` : apresentação em écran. Use `print` para criar versões imprimíveis.

`panelright` coloca um painel de navegação do lado direito do écran. Se

desejar o painel do lado esquerdo, use `panelleft`.

`french` ou qualquer outra língua suportada, irá mostrar o texto dos botões de navegação de forma apropriada. Esta opção é independente das opções colocadas no pacote babel.

`chocolate` esquema de cor para o painel de navegação. Outras escolhas podem ser `gray`, `orange`, `palegreen`, `bluelace` e `blue`, que é a por omissão.

Depois configura o formato do écran. Como a escala da apresentação irá sempre mudar de acordo com o tamanho real do écran, isto pode ser usado para configurar na generalidade o tamanho das letras:

`\panelwidth` define a largura do painel de navegação;

`\screensize{`***largura***`}{`***altura***`}` define a largura e a altura do écran, incluindo o painel de navegação;

`\marginsize{`***esquerda***`}{`***direita***`}{`***topo***`}{`***fundo***`}` define as margens do documento. No exemplo o documento não está centrado porque os números de secção são mantidos na margem esquerda.

É possível usar uma imagem de fundo, em qualquer um dos formatos suportados pelo pdfTEX usando o comando

`\overlay{`*imagem*`}`

ou se preferir um fundo plano, pode definir a cor usando

`\background{`*cor*`}`

Finalmente, se deseja colocar um logotipo da sua organização no painel de navegação use o comando

`\emblema{`*logotipo*`}`

Se acredita no poder de uma apresentação com exposição sucessiva de pontos de uma lista, pode desejar usar o pacote `pause`. Este, providencia o comando `\pause` que pode ser colocado directamente após o seu texto, exactamente onde quer que o Acrobat faça uma pausa ao mostrar o documento. Este pacote faz parte do sistema `ppower4` (P^4: *Pdf Presentation Post-Processor*), que pós-processa o documento pdf resultante do pdfTEX e o faz dançar, cantar e pedir por comida. Pode tratar o resultado do pdfTEX

passando-o pelo pós-processador. Na linha de comando, faria qualquer coisa como:

```
ppower4 xy.pdf xyz.pdf
```

Para controlar o que aparece em cada slide, pode usar o ambiente `\begin{slide}` ...`\end{slide}`. O conteúdo de cada um irá ser mostrado centrado verticalmente na sua página.

Se compilar o exemplo anterior irá obter uma mensagem de erro:

```
! pdfTeX warning (dest): name{contents} has been
  referenced but does not exist, replaced by a fixed one
```

Isto acontece porque existe um botão no painel de navegação que quer apontar para a tabela de conteúdos e, como este exemplo não contém o comando `\tableofcontents` a resolução da ligação falha.

Se quer que a tabela de conteúdos seja mostrada directamente dentro do painel de navegação, pode usar a opção `paneltoc` ao chamar o pacote pdfscreen. Isto produzirá resultados agradáveis apenas se a sua apresentação tiver poucas e curtas entradas na tabela de conteúdos. Poderá, também, providenciar pequenos títulos para as secções, tal como o faz num documento normal.

Esta pequena introdução só arranha a superfície do que é possível com o pacote pdfscreen e o sistema `PPower4`. Ambos contêm os seus próprios manuais.

4.9 XY-pic

By Alberto Manuel Brandão Simões <albie@alfarrabio.di.uminho.pt>

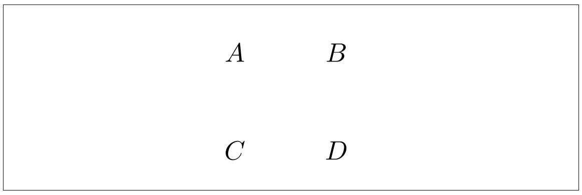
xy é um pacote especial para desenhar diagramas. Para o usar, adicione a seguinte linha ao preâmbulo do seu documento:

```
\usepackage[opções]{xy}
```

onde *opções* é a lista de funções do XY-pic que quer usar. Estas opções são especialmente úteis para encontrar erros no pacote. Pessoalmente, recomendo a opção `all` que indica ao LaTeX para carregar todos os comandos disponíveis no XY.

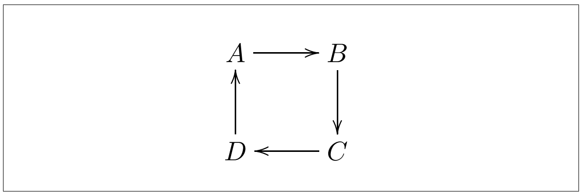
Os diagramas XY-pic são desenhados numa tela orientada à matriz, onde cada elemento do diagrama é colocado numa das posições da matriz:

```
\begin{displaymath}
\xymatrix{A & B \\
          C & D }
\end{displaymath}
```



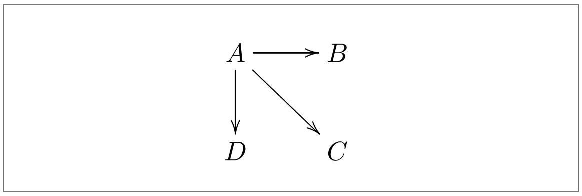
O comando `\xymatrix` deve ser usado sempre em modo matemático. Aqui, especificamos duas linhas e duas colunas. Para converter esta matriz num diagrama podemos adicionar algumas linhas direccionais usando o comando `\ar`.

```
\begin{displaymath}
\xymatrix{ A \ar[r] & B \ar[d] \\
           D \ar[u] & C \ar[l] }
\end{displaymath}
```



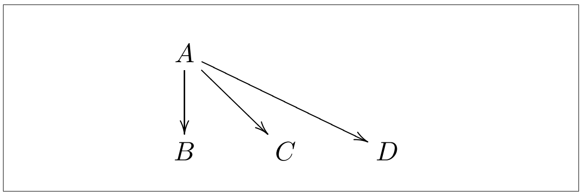
O comando para desenhar a seta é colocado na célula de origem da seta. O argumento é a direcção para a qual a seta deve apontar: esquerda (`left`), direita (`right`), cima (`up`) ou para baixo (`down`).

```
\begin{displaymath}
\xymatrix{
  A \ar[d] \ar[dr] \ar[r] & B \\
  D                       & C }
\end{displaymath}
```



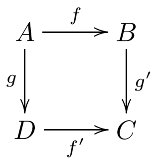
Para criar diagonais, junte mais do que uma direcção. De facto, até pode repetir direcções para criar setas mais compridas.

```
\begin{displaymath}
\xymatrix{
  A \ar[d] \ar[dr] \ar[drr] & & \\
  B                       & C & D }
\end{displaymath}
```



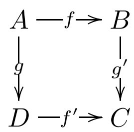
Podemos desenhar diagramas ainda mais interessantes adicionando etiquetas às setas. Para isto, usamos os operadores habituais para expoentes e índices.

```
\begin{displaymath}
\xymatrix{
  A \ar[r]^f \ar[d]_g &
            B \ar[d]^{g'} \\
  D \ar[r]_{f'}       & C }
\end{displaymath}
```



Como mostrado, usa estes operadores como em modo matemático. A única diferença é que o expoente significa "em cima da seta," e índice significa "por baixo da seta." Existe ainda um terceiro operador, a barra vertical: `|` que coloca o texto *dentro* da seta.

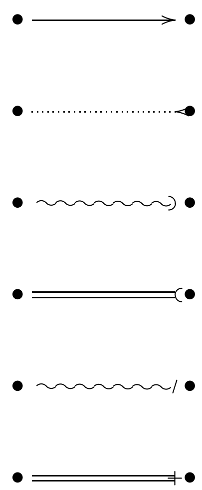
```
\begin{displaymath}
\xymatrix{
  A \ar[r]|f \ar[d]|g &
            B \ar[d]|{g'} \\
  D \ar[r]|{f'}       & C }
\end{displaymath}
```



Para desenhar uma seta com um buraco, use `\ar[...]|\hole`.

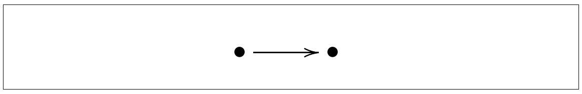
Em algumas situações, é importante distinguir entre diferentes tipos de setas. Isto pode ser feito colocando-lhe etiquetas, ou mudando a sua aparência:

```
\begin{displaymath}
\xymatrix{
 \bullet\ar@{->}[rr] && \bullet\\
 \bullet\ar@{.<}[rr] && \bullet\\
 \bullet\ar@{~)}[rr] && \bullet\\
 \bullet\ar@{=(}[rr] && \bullet\\
 \bullet\ar@{~/}[rr]  && \bullet\\
 \bullet\ar@{=+}[rr]   && \bullet
}
\end{displaymath}
```

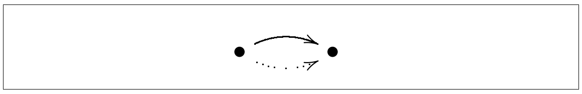


Note a diferença entre os seguintes dois diagramas:

```
\begin{displaymath}
\xymatrix{
 \bullet \ar[r]
         \ar@{.>}[r] &
 \bullet
}
\end{displaymath}
```



```
\begin{displaymath}
\xymatrix{
 \bullet \ar@/^/[r]
         \ar@/_/@{.>}[r] &
 \bullet
}
\end{displaymath}
```



Os modificadores entre as barras define a forma como as curvas são desenhadas. O XY-pic oferece muitas mais formas de influenciar o desenho das curvas; para mais informação, veja a documentação e o tutorial de XY-pic.

Capítulo 5

Configurar o LaTeX

Os documentos produzidos usando os comandos apresentados até aqui parecerão aceitáveis a uma grande audiência. Não seguem um estilo muito trabalhado, mas obedecem às regras estabelecidas como correctas para um bom documento, e que farão o documento agradável e agradável de ler.

No entanto, existem situações onde o LaTeX não dispõe de comandos ou ambientes que satisfaçam as suas necessidades, ou o resultado produzido por um comando já existente não está de acordo com os seus requisitos.

Este capítulo tentará dar algumas ideias sobre como ensinar novos truques ao LaTeX e como fazê-lo produzir resultados que são diferentes aos disponíveis de raíz.

5.1 Novos Comandos, Ambientes e Pacotes

Deve ter reparado que todos os comandos que introduzi neste livro são apresentados numa caixa, e que aparecem no índice no fim do livro. Em vez de usar directamente os comandos LaTeX necessários para obter este resultado, criei um pacote no qual defini novos comandos e ambientes para este fim. Agora, escrevo simplesmente:

```
\begin{lscommand}
\ci{dum}
\end{lscommand}
```

\dum

Neste exemplo estou a utilizar quer um novo ambiente chamado `lscommand`, que é responsável por desenhar a caixa à volta do comando, e um novo comando denominado `\ci` que escreve o nome do comando e também coloca a entrada correspondente no índice. Pode verificar isto olhando para o comando `\dum` no índice no fim deste livro, onde aparecerá uma entrada para `\dum`, apontando cada uma das páginas onde mencionei o comando `\dum`.

Se decidir que já não gosto que os comandos sejam escritos numa caixa, posso alterar simplesmente a definição do ambiente `lscommand` para criar uma nova aparência. Isto é muito mais fácil do que andar por todo o documento à caça de todos os lugares onde usei alguns comandos genéricos LaTeX para desenhar uma caixa à volta de algumas palavras.

5.1.1 Novos Comandos

Para adicionar os seus próprios comandos, use o comando

`\newcommand{`*nome*`}[`*num*`]{`*definição*`}`

Basicamente, o comando necessita de dois argumentos: o *nome* do comando que quer criar, e a *definição* do comando. O argumento *num* em parêntesis rectos é opcional e especifica o número de argumentos que o novo comando recebe (são possíveis até 9). Se não especificar, o valor utilizado é 0, ou seja, nenhum argumento é permitido.

Os dois exemplos seguintes devem ajudar a apanhar a ideia. O primeiro exemplo define um novo comando chamado `\npil`. Este é um atalho para "A Não Tão Pequena Introdução ao LaTeX 2_ε." Um comando deste género pode tornar-se útil se precisa de escrever o título deste livro muitas e muitas vezes.

```
\newcommand{\npil}{A Não
    Tão Pequena Introdução ao
    \LaTeXe}
Esta é a  ‘‘\npil’’ \ldots{}
‘‘\npil’’
```

Esta é a "A Não Tão Pequena Introdução ao LaTeX 2_ε" … "A Não Tão Pequena Introdução ao LaTeX 2_ε"

O próximo exemplo ilustra a definição de um novo comando que recebe um argumento. A etiqueta `#1` vai ser substituída pelo argumento que especificar. Se quiser usar mais do que um argumento, use `#2` e assim sucessivamente.

```
\newcommand{\txsit}[1]
 {Esta é a \emph{#1} Pequena
      Introdução ao \LaTeXe}
% no corpo do documento:
\begin{itemize}
\item \txsit{Não Tão}
\item \txsit{Muito}
\end{itemize}
```

- Esta é a *Não Tão* Pequena Introdução ao LaTeX 2_ε
- Esta é a *Muito* Pequena Introdução ao LaTeX 2_ε

LaTeX não permite que crie um novo comando que substitua um já existente. Mas, existe um comando especial no caso de querer fazer isto. Nesse caso, use explicitamente o comando `\renewcommand` que funciona da mesma forma que o comando `\newcommand`.

Em alguns casos, pode querer usar o comando `\providecommand` que funciona como o `\newcommand`, mas se o comando já existir, o LaTeX irá ignora-lo silenciosamente.

Existem alguns pontos a tomar em conta quando existem espaços após comandos LaTeX. Veja a página 5 para mais informação.

5.1.2 Novos Ambientes

Tal como com o comando `\newcommand`, existe também um comando para criar os seus próprios ambientes. O comando `\newenvironment` usa a seguinte sintaxe:

`\newenvironment{`*nome*`}[`*num*`]{`*antes*`}{`*depois*`}`

De novo, o comando `\newcommand` pode usar `\newenvironment` com um argumento opcional. O material especificado no argumento *antes*, é processado antes do texto incluso no ambiente seja processado. O conteúdo do argumento *depois* é processado quando o comando `\end{`*nome*`}` é encontrado.

O seguinte exemplo ilustra a utilização do comando `\newenvironment`.

```
\newenvironment{king}
 {\rule{1ex}{1ex}%
      \hspace{\stretch{1}}}
 {\hspace{\stretch{1}}%
      \rule{1ex}{1ex}}

\begin{king}
Os meus pensamentos \ldots
\end{king}
```

■	Os meus pensamentos . . .	■

O argumento *num* é usado da mesma forma que o do `\newcommand`. O LaTeX também não permite que defina um ambiente que já exista. Se por alguma razão quiser mudar um ambiente já existente, pode fazê-lo com o comando `\renewenvironment` que usa a mesma sintaxe do comando `\newenvironment`.

O comando usado neste exemplo irá ser explicado mais tarde: Para o comando `\rule` veja a página 103, para `\stretch` vá à página 97, e mais informação sobre `\hspace` pode ser encontrada na página 97.

5.1.3 O Seu Próprio Pacote

Se definir um grande número de novos comandos e ambientes, o preâmbulo do seu documento ficará bastante longo. Nesta situação, é a boa ideia criar

um pacote LaTeX contendo todas as suas definições de comandos e ambientes. Depois pode usar o comando `\usapackage` para tornar as definições disponíveis no seu documento.

```
% Pacote de Demonstração por Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\npil}{A não tão Pequena Introdução ao \LaTeXe}
\newcommand{\txsit}[1]{A \emph{#1} Tão
                       Introdução ao \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

Figura 5.1: Pacote de Exemplo.

Escrever um pacote consiste basicamente em copiar o conteúdo do preâmbulo do seu documento para um ficheiro separado com um nome com a extensão `.sty`. Existe um comando especial

`\ProvidesPackage{`*nome do pacote*`}`

para usar no topo do seu pacote. O comando `\ProvidesPackage` indica ao LaTeX o nome do pacote e irá permitir que apresente mensagens de erro quando tentar incluir mais do que uma vez um pacote. A figura 5.1 mostra um pequeno exemplo de um pacote que contém os comandos definidos nos exemplos anteriores.

5.2 Letras e Tamanhos

5.2.1 Tipos de letra

O LaTeX escolhe os tipos de letra e respectivos tamanhos apropriados baseando-se na estrutura lógica do documento (secções, notas de rodapé, …). Em alguns casos, pode desejar mudar os tipos ou tamanho de letras à mão. Para fazer isto, pode usar os comandos listados nas tabelas 5.1 e 5.2. O tamanho de cada tipo de letra é uma definição que depende na classe de documento e nas suas opções. A tabela 5.3 mostra o tamanho absoluto em pontos para estes comandos como implementados nas classes standard.

```
{\small Os pequenos e
\textbf{gordos} Romanos mandaram}
{\Large em toda a grande
\textit{Itália}.}
```

Os pequenos e **gordos** Romanos mandaram em toda a grande *Itália*.

Uma propriedade importante do LaTeX 2_ε é que os atributos de letras são independentes. Isto significa que pode mandar alterar o tipo ou tamanho

de letra e, no entanto, manter os atributos de *bold* ou *itálico* que tinha feito anteriormente.

Em *modo matemático* pode usar os *comandos* de mudança de letra para sair temporariamente do *modo matemático* e entrar em texto normal. Se precisar de mudar para outro tipo de letra para escrever matemática, existe outro conjunto de comandos especiais: consulte a tabela 5.4.

Em ligação com os comandos de mudança de tamanho, as chavetas têm um papel bastante importante. São usados para construir *grupos*. Os grupos limitam a zona de quase todos os comandos LaTeX.

```
Ele gosta de letras {\LARGE grandes e
{\small pequenas}}.
```

Ele gosta de letras grandes e pequenas.

Os comandos de mudança de tamanho de letra também mudam o espaçamento entre linhas, mas apenas se o parágrafo acaba dentro do alcance do comando de mudança de tamanho. A chaveta a fechar `}` não deve, portanto, aparecer cedo demais. Note a posição do comando `\par` nos dois exemplos seguintes. [1]

[1] `\par` é equivalente a uma linha em branco

Tabela 5.1: Letras.

`\textrm{...}`	romano	`\textsf{...}`	sans serif
`\texttt{...}`	`à maquina`		
`\textmd{...}`	médio	`\textbf{...}`	**tipo gordo**
`\textup{...}`	em pé	`\textit{...}`	*itálico*
`\textsl{...}`	*para a frente*	`\textsc{...}`	PEQUENAS MAIÚSCULAS
`\emph{...}`	*salientado*	`\textnormal{...}`	texto normal

Tabela 5.2: Tamanho de Letra.

`\tiny`	letra minúscula	`\Large`	letras maiores
`\scriptsize`	letra muito pequena	`\LARGE`	letras muito grandes
`\footnotesize`	letra bastante pequena	`\huge`	letras enormes
`\small`	letra pequena	`\Huge`	as maiores
`\normalsize`	tamanho normal		
`\large`	letras largas		

Tabela 5.3: Tamanhos Absolutos nas Classes Padrão.

tamanho	10pt (omissão)	opção 11pt	opção 12pt
`\tiny`	5pt	6pt	6pt
`\scriptsize`	7pt	8pt	8pt
`\footnotesize`	8pt	9pt	10pt
`\small`	9pt	10pt	11pt
`\normalsize`	10pt	11pt	12pt
`\large`	12pt	12pt	14pt
`\Large`	14pt	14pt	17pt
`\LARGE`	17pt	17pt	20pt
`\huge`	20pt	20pt	25pt
`\Huge`	25pt	25pt	25pt

Tabela 5.4: Letras Matemáticas.

Comando	*Exemplo*	*Resultado*
`\mathcal{...}`	`$\mathcal{B}=c$`	$\mathcal{B}=c$
`\mathrm{...}`	`$\mathrm{K}_2$`	K_2
`\mathbf{...}`	`$\sum x=\mathbf{v}$`	$\sum x=\mathbf{v}$
`\mathsf{...}`	`$\mathsf{G\times R}$`	$\mathsf{G\times R}$
`\mathtt{...}`	`$\mathtt{L}(b,c)$`	$\mathtt{L}(b,c)$
`\mathnormal{...}`	`$\mathnormal{R_{19}}\neq R_{19}$`	$\mathnormal{R_{19}}\neq R_{19}$
`\mathit{...}`	`$\mathit{ffi}\neq ffi$`	$\mathit{ffi}\neq ffi$

```
{\Large Não leia isto! Isto não é
verdade. Tem de acreditar
em mim!\par}
```

Não leia isto! Isto não é verdade. Tem de acreditar em mim!

```
{\Large Isto também não é verdade.
Mas lembre-se que sou
um mentiroso.}\par
```

Isto também não é verdade. Mas lembre-se que sou um mentiroso.

Se quer activar o comando de mudança de tamanho de letra para um parágrafo inteiro de texto ou mesmo mais, deve usar a sintaxe de ambiente para estes comandos.

```
\begin{Large}
Isto não é verdade.
Mas mais uma vez, o que são
estes dias \ldots
\end{Large}
```

Isto não é verdade. Mas mais uma vez, o que são estes dias . . .

Isto irá facilitar a contagem das muitas chavetas.

5.2.2 Perigo, Will Robinson, Perigo

Como foi referido no início deste capítulo, é perigoso infestar os seus documentos com comandos explícitos como estes, porque funcionam em oposição à ideia básica do LaTeX, que é separar a estrutura lógica e visual dos documentos. Isto significa que se usar o mesmo comando de mudança de letra em vários sítios de forma a desenhar um tipo especial de informação, deve usar o `\newcommand` para definir um "comando lógico" para o comando de mudança de letras.

```
\newcommand{\oops}[1]{\textbf{#1}}
Não \oops{entre} neste quarto,
está ocupado por uma \oops{máquina}
de origem e objectivos desconhecidos.
```

Não **entre** neste quarto, está ocupado por uma **máquina** de origem e objectivos desconhecidos.

Esta abordagem tem a vantagem de que pode decidir mais tarde se quer utilizar uma outra representação visual do perigo sem ser o `\textbf` sem ter de alterar todo o seu documento, identificando as ocorrências de `\textbf` e descobrindo, para cada uma, de é ou não um dos casos em que está a apontar perigo, ou se foi usado por qualquer outra razão.

5.2.3 Aviso

Para concluir esta jornada na terra das letras, aqui está uma pequena palavra de aviso:

> **Lembre-se!** *QUANTOS* MA IS tipos de letra VOCê usar **num** `documento`, *o* mais LEGÍVEL e *bonito ele ficara!*

5.3 Espaçamento

5.3.1 Espaço entre linhas

Se quer usar um espaço maior entre linhas num documento, pode alterar o seu valor colocando o comando

```
\linespread{factor}
```

no preâmbulo do seu documento. Use `\linespread{1.3}` para um espaçamento de "um e meio", e `\linespread{1.6}` para um espaçamento "duplo". Normalmente as linhas não estão espalhadas, pelo que o factor por omissão é 1.

5.3.2 Formatação de Parágrafos

No LaTeX, existem dois parâmetros que influenciam o formato dos parágrafos. Ao colocar uma definição como

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

no preâmbulo do seu documento, pode alterar o formado dos parágrafos. Estes dois comandos aumentam o espaço entre dois parágrafos colocando a indentação a zero.

As partes `plus` e `minus` do comprimento acima instrui o TeXde que pode comprimir e expandir o espaçamento entre parágrafos pela quantidade especificada se for necessário para colocar os parágrafos de forma correcta na página.

Na Europa continental, os parágrafos são, normalmente, separados por algum espaço a mais e não indentados. Mas, cuidado, isto também afecta a tabela de conteúdos. As suas linhas irão aparecer mais espaçadas. Para isto não acontecer, pode mover os dois comandos do preâmbulo até a um ponto do seu documento após o `\tableofcontents` ou, simplesmente, não os usar de todo, porque irá reparar que a maior parte dos livros profissionais usam indentação e não espaço para separar os parágrafos.

Se quer indentar um parágrafo que não o está, pode usar o comando

```
\indent
```

no início do parágrafo.[2] Obviamente, isto irá apenas afectar o texto quando o `\parindent` não está a zero.

Para criar um parágrafo não indentado, pode usar

```
\noindent
```

como o primeiro comando do parágrafo. Isto pode ser útil quando começa um documento com texto e não com um comando que seccione o documento.

5.3.3 Espaço Horizontal

O LaTeX determina os espaços entre palavras e frases automaticamente. Para adicionar espaço adicional, use:

\hspace{*comprimento*}

Se um destes espaços deve ser mantido mesmo que atinja o fim ou início duma linha, use `\hspace*` em vez de `\hspace`. O *comprimento* é, no caso mais simples, apenas um número e uma unidade. As unidades mais importantes estão listadas na tabela 5.5.

```
Este\hspace{1.5cm}é um espaço
de 1.5 cm.
```

Este   é um espaço de 1.5 cm.

O comando

\stretch{*n*}

gera um espaço especial. Ele estica até que todo o espaço restante na linha fique completo. Se dois comandos `\hspace{\stretch{`*n*`}}` forem invocados na mesma linha, irão crescer de acordo com o factor indicado.

```
x\hspace{\stretch{1}}
x\hspace{\stretch{3}}x
```

x   x   x

Quando se usa espaço horizontal juntamente com texto, pode fazer sentido obrigar o espaço a ajustar-se relativamente ao tamanho actual do tipo de letra. Pode fazer isto usando unidades relativas ao texto: `em` e `en`.

[2] Para indentar o primeiro parágrafo após cada título de secção, use o pacote indentfirstq que vem no conjunto de pacotes ‘tools’

Tabela 5.5: Unidades do TEX.

`mm`	milímetro ≈ 1/25 polegadas	⊔
`cm`	centímetro = 10 mm	⊔___⊔
`in`	polegada = 25.4 mm	⊔_______⊔
`pt`	ponto ≈ 1/72 polegada ≈ $\frac{1}{3}$ mm	⊔
`em`	aprox largura de um 'M' no tipo de letra actual	⊔_⊔
`ex`	aprox altura de um 'x' no tipo de letra actual	⊔

```
{\Large{}gran\hspace{1em}de}\\
{\tiny{}peque\hspace{1em}no}
```

gran   de
peque no

5.3.4 Espaço Vertical

O espaço entre parágrafos, secções, subsecções, ... é determinado automaticamente pelo LaTeX. Se necessário, pode adicionar espaço vertical adicional *entre dois parágrafos* com o comando:

`\vspace{`*comprimento*`}`

Este comando deve ser normalmente usado entre duas linhas vazias. Se o espaço deve ser preservado no inicio ou no fim de uma página, utilize a versão estrelada do comando: `\vspace*` em vez de `\vspace`.

O comando `\stretch` em conexão com o comando `\pagebreak` pode ser usado para imprimir texto na última linha de uma página, ou para centrar texto verticalmente numa página.

```
Algum texto \ldots

\vspace{\stretch{1}}
Isto aparece na última linha da página.\pagebreak
```

Espaço adicional entre duas linhas do *mesmo* parágrafo ou dentro de uma tabela é especificado com o comando

`\\[`*comprimento*`]`

Com `\bigskip` e `\smallskip` pode saltar uma quantidade pré definida de espaço vertical sem ter de se preocupar com os números exactos.

5.4 Formato da Página

O LaTeX 2_ε permite especificar o tamanho do papel no comando `\documentclass`. Depois, automaticamente selecciona as margens de texto correctas. Mas, por vezes, pode não estar contente com os valores pré-definidos. Naturalmente, pode os alterar. A figura 5.2 mostra todos os parâmetros que podem ser alterados. A figura foi produzida com o pacote layout do conjunto 'tools'.[3]

ESPERE! . . . antes de se lançar freneticamente a "Toca a fazer as páginas estreitas um pouco mais largas", tire alguns segundos para pensar. Como na maioria dos casos, existe uma boa razão para que o formato da página seja o que é.

Claro, comparado com a sua página tirada do MS Word, parece muito mais estreita. Mas dê uma olhadela ao seu livro favorito[4] e conte o número de caracteres numa linha de texto normal. Irá notar que não existem mais do que 66 caracteres em cada linha. Agora, faça o mesmo nas páginas do seu documento LaTeX. A experiência mostra que a leitura se torna difícil assim que existem mais caracteres numa única linha. Isto é porque é difícil aos olhos mover do fim de uma linha para o inicio da próxima. Esta também é a razão pela qual os jornais são escritos em múltiplas colunas.

Então, se aumentar a largura do texto do seu livro, lembre-se que está a tornar a vida mais difícil aos seus leitores. Mas chega de avisos, prometi que dizia como se faz isso . . .

O LaTeX dispõe de dois comandos para mudar estes parâmetros. São usados normalmente no preâmbulo do documento.

O primeiro comando atribuí um valor fixo a qualquer um dos parâmetros:

`\setlength{`*parâmetro*`}{`*comprimento*`}`

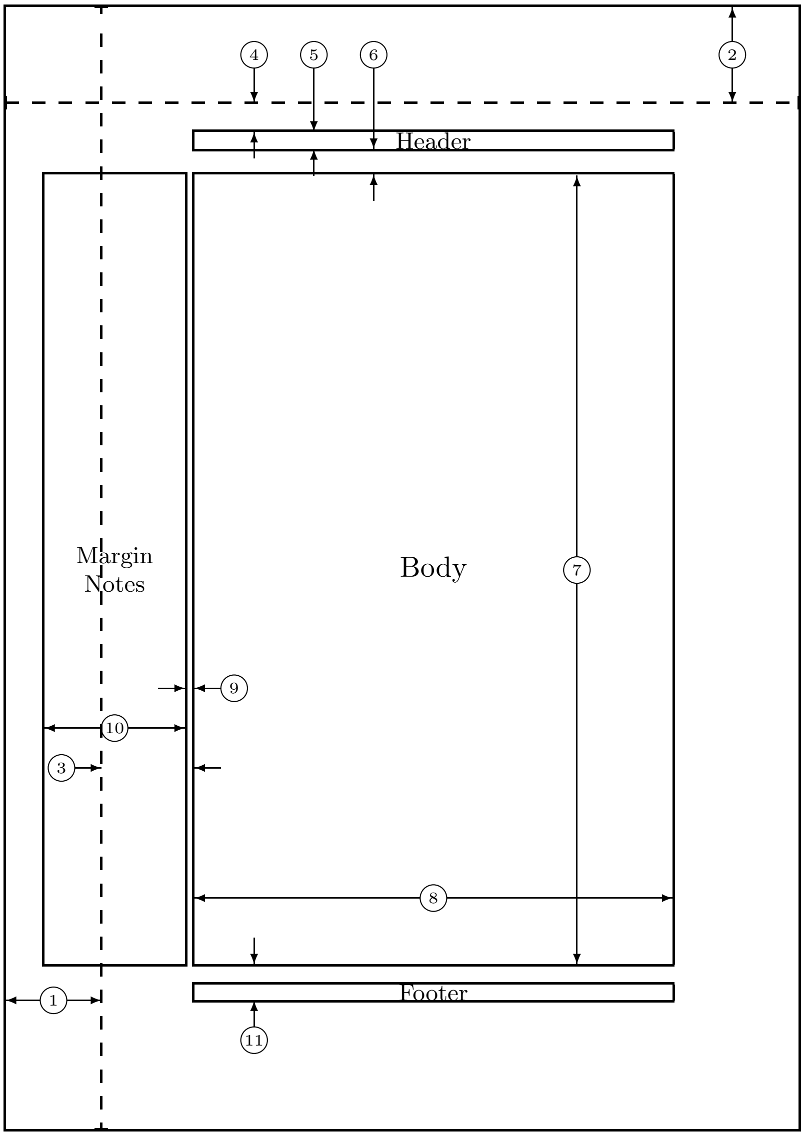
O segundo comando adiciona um valor a qualquer um dos parâmetros:

`\addtolength{`*parâmetro*`}{`*comprimento*`}`

Este segundo comando é, normalmente, mais útil que o `\setlength` porque pode funcionar de uma forma relativa aos valores já existentes. Para adicionar um centímetro para a largura normal do texto, utilize o comando seguinte no preâmbulo do seu documento:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

[3] `CTAN:/tex-archive/macros/latex/required/tools`
[4] Quero dizer, um livro realmente produzido por uma editora de reputação.



1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 22pt or \evensidemargin	4	\topmargin = 22pt
5	\headheight = 13pt	6	\headsep = 19pt
7	\textheight = 595pt	8	\textwidth = 360pt
9	\marginparsep = 7pt	10	\marginparwidth = 106pt
11	\footskip = 27pt		\marginparpush = 5pt (not shown)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

Figura 5.2: Parâmetros do Formato de Página.

Neste contexto, pode querer olhar para o pacote calc, que permite usar operações aritméticas no argumento de este e de outros comandos onde deveria introduzir valores.

5.5 Mais divertimento com comprimentos

Sempre que possível, tento não usar comprimentos absolutos em documentos LaTeX. Prefiro tentar basear as coisas na largura ou altura de outros elementos das páginas. Para a largura de uma figura, podia relacionar com `\textwidth` de forma a que ela coubesse numa página.

Os seguintes 3 comandos permitem determinar a largura, altura e profundidade de um texto.

```
\settoheight{variável}{texto}
\settodepth{variável}{texto}
\settowidth{variável}{texto}
```

O seguinte exemplo mostra uma possível aplicação para estes comandos.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[0pt][r]{#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Onde}$a$,
$b$ -- são adjuntos do ângulo
recto de um triângulo rectângulo.

$c$ -- é a hipotenusa do
triângulo e sente-se sozinha.

$d$ -- finalmente, nem sequer
aparece. Não é curioso?
\end{vardesc}
```

$$a^2 + b^2 = c^2$$

Onde: a, b – são adjuntos do ângulo recto de um triângulo rectângulo.

c – é a hipotenusa do triângulo e sente-se sozinha.

d – finalmente, nem sequer aparece. Não é curioso?

5.6 Caixas

O LaTeX constrói as suas páginas movendo caixas. A principio, cada letra é uma pequena caixa, que é depois colada a outras letras para formar palavras. Estas são de novo coladas a outras palavras, mas com cola especial, que é

elástica e portanto uma série de palavras pode ser encolhida ou esticada para preencher exactamente uma linha de texto na página.

Admito, esta é uma versão muito simplicista do que realmente acontece, mas na verdade, o TeX opera com cola e caixas. Não só uma letra que pode ser uma caixa. Pode colocar virtualmente tudo numa caixa, incluindo outras caixas. Cada caixa irá depois ser manuseada pelo LaTeX como se fosse uma simples letra.

Nos capítulos anteriores, já encontrou algumas caixas, no entanto, não lhe disse. O ambiente `tabular` e o `\includegraphics`, por exemplo, produzem ambos uma caixa. Isto significa que pode colocar facilmente duas tabelas ou imagens lado a lado. Só tem de ter a certeza que a sua largura não é mais larga que a largura do texto.

Também pode empacotar um parágrafo da sua escolha numa caixa com o comando

`\parbox[`*pos*`]{`*largura*`}{`*texto*`}`

ou com o ambiente

`\begin{minipage}[`*pos*`]{`*largura*`}` texto `\end{minipage}`

O parâmetro `pos` pode tomar uma das letras `c`, `t` ou `b` para controlar o alinhamento vertical da caixa, relativamente à linha base do texto circundante. A `largura` toma um valor que especifica a largura da caixa. A principal diferença entre `minipage` e `\parbox` é que não pode usar todos os comandos e ambientes dentro de uma `parbox` enquanto quase tudo é possível numa `minipage`.

Enquanto `\parbox` empacota um parágrafo fazendo quebras de linha e tudo o mais, existe também uma classe de comandos para caixotes que operam apenas em material alinhado horizontalmente. Já conhecemos um deles. É chamado `\mbox`, e empacota simplesmente uma série de caixas dentro de uma outra, e pode ser usado para prevenir a hifenização de palavras. Como pode colocar caixas dentro de qualquer caixa, estes empacotadores horizontais dão-lhe uma flexibilidade ilimitada.

`\makebox[`*largura*`][`*pos*`]{`*texto*`}`

A `largura` define a largura da caixa resultante vista do lado de fora.[5] Além do comprimento das expressões, pode também usar `\width`, `\height`, `\depth` e `\totalheight` no parâmetro de largura. Todos eles são valores obtidos medindo o *texto* escrito. O parâmetro *pos* toma um valor de entre

[5] Isto significa que pode ser mais pequena do que o material lá dentro. Pode até colocar a largura a 0pt de forma a que o texto dentro da caixa irá ser escrita sem influenciar as caixas circundantes.

as letras: **c**entro, à esquerda (**l**eft), à direita (**r**ight) ou **s** que espalha o texto dentro da caixa para a preencher.

O comando `\framebox` funciona exactamente da mesma forma que `\makebox`, mas desenha uma caixa à volta do texto.

O seguinte exemplo mostra algumas coisas que pode fazer com os comandos `\makebox` e `\framebox`.

```
\makebox[\textwidth]{%
    c e n t r a d o}\par
\makebox[\textwidth][s]{%
    e s p a l h a d o}\par
\framebox[1.1\width]{Agora
    estou encaixilhado!} \par
\framebox[0.8\width][r]{Bolas,
    estou tão largo} \par
\framebox[1cm][l]{esquece,
    Eu sou assim}
Consegue ler isto?
```

c e n t r a d o

e s p a l h a d o

Agora estou encaixilhado!

Bolas, estou tão largo

esquece, Eu sou assim Consegue ler isto?

Agora que controlamos a horizontal, o próximo passo obvio é ir para o vertical.[6] Sem problemas para o LaTeX. O comando

`\raisebox{`*içar*`}[`*prof*`][`*altura*`]{`*texto*`}`

permite definir as propriedades verticais de uma caixa. Pode usar o `\width`, `\height`, `\depth` e `\totalheight` nos primeiros três parâmetros, de forma a actual de acordo com o tamanho da caixa dentro do argumento *texto*.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}}
ele gritou mas nem sequer o mais
próximo notou que alguma coisa
terrível lhe tinha acontecido.
```

Aaaaaaargh ele gritou mas nem sequer o mais próximo notou que alguma coisa terrível lhe tinha acontecido.

5.7 Réguas e Estruturas

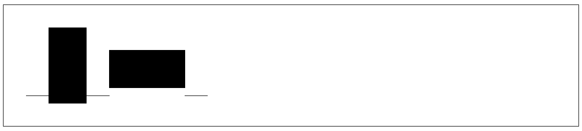
Algumas páginas atrás deve ter reparado no comando

`\rule[`*içar*`]{`*largura*`}{`*altura*`}`

Normalmente ele produz uma simples caixa preta.

[6] O controlo total é obtido apenas controlando a horizontal tão bem como a vertical.... . .

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



Isto é útil para desenhar linhas horizontais e verticais. A linha na página de título, por exemplo, foi criada com um comando `\rule`.

Um caso especial é uma régua sem largura mas alguma altura. Em tipografia, isto é chamado de estrutura. É usada para garantir que um elemento numa página tem uma determinada altura mínima. Pode usar num ambiente `tabular` para ter a certeza que uma linha tem uma altura mínima.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Pitprop \ldots\\
\hline
\rule{0pt}{4ex}Strut\\
\hline
\end{tabular}
```

Pitprop …
Strut

FIM.

Bibliografia

[1] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.

[2] Donald E. Knuth. *The TeXbook,* Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.

[3] Michel Goossens, Frank Mittelbach and Alexander Samarin. *The LaTeX Companion.* Addison-Wesley, Reading, Massachusetts, 1994, ISBN 0-201-54199-8.

[4] Cada instalação LaTeX deve providenciar um *LaTeX Local Guide* que explica os pormenores que são especiais nesse sistema em particular. Deve estar contido num ficheiro denominado `local.tex`. Infelizmente, muitos operadores de sistema preguiçosos não providenciam este documento. Neste caso, vá e peça ajuda ao seu mestre de LaTeX.

[5] LaTeX3 Project Team. *LaTeX 2ε for authors.* Vem com a distribuição LaTeX 2ε como `usrguide.tex`.

[6] LaTeX3 Project Team. *LaTeX 2ε for Class and Package writers.* Vem com a distribuição do LaTeX 2ε como `clsguide.tex`.

[7] LaTeX3 Project Team. *LaTeX 2ε Font selection.* Vem com a distribuição do LaTeX 2ε como `fntguide.tex`.

[8] D. P. Carlisle. *Packages in the 'graphics' bundle.* Vem com o conjunto de ferramentas 'graphics' como `grfguide.tex`, disponível da mesma origem da sua distribuição de LaTeX.

[9] Rainer Schöpf, Bernd Raichle, Chris Rowley. *A New Implementation of LaTeX's verbatim Environments.* Vem com o conjunto 'tools' como `verbatim.dtx`, disponível da mesma origem da sua distribuição de LaTeX.

[10] Graham Williams. *The TeX Catalogue* é uma listagem bastante completa de muitos pacotes relacionados com o TEX e o LATEX. Disponível na Internet desde `CTAN:/tex-archive/help/Catalogue/catalogue.html`

[11] Keith Reckdahl. *Using EPS Graphics in LATEX 2ε Documents* que explica tudo e muito mais do que alguma vez quis saber sobre ficheiros EPS e o seu uso em documento LATEX. Disponível na Internet desde `CTAN:/tex-archive/info/epslatex.ps`

[12] Kristoffer H. Rose *XY-pic User's Guide*. Disponível no CTAN juntamente com a distribuição do XY-pic.

Índice